

Distributed Message Service for Kafka

API Reference

Issue 14
Date 2025-02-11



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

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

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Before You Start	1
2 API Overview	3
3 Calling APIs	7
3.1 Making an API Request	7
3.2 Authentication	11
3.3 Returned Values	12
4 Getting Started	15
5 APIs V2 (Recommended)	17
5.1 Managing Lifecycle	17
5.1.1 Creating an Instance	17
5.1.2 Listing All Instances	40
5.1.3 Querying an Instance	53
5.1.4 Deleting an Instance	65
5.1.5 Modifying Instance Information	68
5.1.6 Batch Restarting or Deleting Instances	78
5.1.7 Obtaining Instance Configurations	87
5.1.8 Modifying Instance Configurations	92
5.1.9 Upgrading an Instance	97
5.2 Instance Management	99
5.2.1 Resetting the Password	99
5.2.2 Resetting Kafka Manager Password	103
5.2.3 Restarting Kafka Manager	107
5.2.4 Configuring Automatic Topic Creation	110
5.2.5 Modifying the Private IP Address for Cross-VPC Access	114
5.2.6 Querying Kafka Cluster Metadata	119
5.2.7 Querying Coordinator Details of a Kafka Instance	124
5.2.8 Reassigning Replicas of a Topic for a Kafka Instance	128
5.2.9 Modifying Kafka Access Modes	133
5.2.10 Querying the Disk Usage Status of Topics	135
5.2.11 Initiating Partition Reassigning for a Kafka Instance	140
5.2.12 Disabling Kafka Manager	149
5.2.13 Deleting a User or Client Quota	152

5.2.14 Querying User or Client Quotas.....	157
5.2.15 Creating User or Client Quotas.....	162
5.2.16 Modifying User or Client Quotas.....	164
5.3 Smart Connect.....	169
5.3.1 Enabling Smart Connect (Pay-per-Use Instance).....	169
5.3.2 Disabling Smart Connect (Pay-per-Use Instance).....	176
5.3.3 Creating a Smart Connect Task.....	180
5.3.4 Querying Smart Connect Tasks.....	202
5.3.5 Querying Smart Connect Task Details.....	211
5.3.6 Deleting a Smart Connect Task.....	219
5.3.7 Pausing a Smart Connect Task.....	222
5.3.8 Restarting a Smart Connect Task.....	225
5.3.9 Starting a Smart Connect Task or Restarting a Paused or Running Smart Connect Task.....	228
5.4 Specification Modification Management.....	232
5.4.1 Increasing Instance Specifications.....	232
5.4.2 Querying Product Information for Instance Specification Modification.....	243
5.5 Topic Management.....	252
5.5.1 Producing Messages to Kafka.....	252
5.5.2 Creating a Topic for a Kafka Instance.....	257
5.5.3 Listing Topics of a Kafka Instance.....	263
5.5.4 Modifying Topics of a Kafka Instance.....	268
5.5.5 Batch Deleting Topics of a Kafka Instance.....	274
5.5.6 Querying the Partition List of a Topic.....	279
5.5.7 Querying the Current Producer List of a Topic.....	283
5.5.8 Querying Topic Details.....	287
5.6 Managing Consumer Groups.....	293
5.6.1 Querying Consumer Group Details.....	293
5.6.2 Querying All Consumer Groups.....	299
5.6.3 Deleting Consumer Groups of a Kafka Instance in Batches.....	303
5.6.4 Creating a Consumer Group.....	308
5.6.5 Resetting Consumer Group Offset to the Specified Position.....	312
5.6.6 Querying the Offset of a Consumer Group.....	319
5.6.7 Modifying All Consumer Groups.....	321
5.6.8 Querying a Specified Consumer Group.....	322
5.6.9 Deleting a Specified Consumer Group.....	326
5.6.10 Modifying a Specified Consumer Group.....	327
5.6.11 Querying Topics of a Specified Consumer Group.....	331
5.6.12 Querying Consumers in a Specified Consumer Group.....	333
5.7 User Management.....	335
5.7.1 Querying the User List.....	335
5.7.2 Creating a User.....	339
5.7.3 Deleting Users in Batches.....	344

5.7.4 Resetting a User Password.....	348
5.7.5 Modifying User Parameters.....	352
5.7.6 Querying User Permissions.....	357
5.7.7 Granting User Permissions.....	361
5.8 Managing Messages.....	367
5.8.1 Querying Messages.....	367
5.8.2 Querying a Message with a Specified Offset.....	373
5.8.3 Querying a Message with a Specified Time Period.....	377
5.8.4 Querying Offset of the Earliest Message in a Partition.....	382
5.8.5 Querying Offset of the Latest Message in a Partition.....	386
5.8.6 Deleting a Kafka Message.....	389
5.9 Background Task Management.....	391
5.9.1 Listing Background Tasks.....	392
5.9.2 Querying a Background Task.....	396
5.9.3 Deleting a Background Task.....	400
5.10 Tag Management.....	403
5.10.1 Batch Adding or Deleting Tags.....	404
5.10.2 Listing Tags of an Instance.....	409
5.10.3 Listing Tags of a Project.....	413
5.11 Diagnosis Management.....	416
5.11.1 Pre-check Before Diagnosing Message Stack.....	416
5.11.2 Creating a Message Stack Diagnosis Task.....	421
5.11.3 Querying the Message Stack Diagnosis Report List.....	425
5.11.4 Batch Deleting Message Stack Diagnosis Reports.....	429
5.11.5 Querying Diagnosis Report Details.....	433
5.12 Other APIs.....	439
5.12.1 Listing Maintenance Time Windows.....	439
5.12.2 Listing AZ Information.....	443
5.12.3 Querying Product Specifications List.....	447
5.12.4 Querying Kafka Instance Monitoring Dimensions.....	455
5.12.5 Querying vCPUs of a Kafka Flavor.....	461
6 Permissions and Supported Actions.....	463
7 Out-of-Date APIs.....	471
7.1 API V1.....	471
7.1.1 APIs for Managing Instances.....	471
7.1.1.1 Creating an Instance.....	471
7.1.1.2 Querying an Instance.....	479
7.1.1.3 Modifying an Instance.....	485
7.1.1.4 Deleting an Instance.....	488
7.1.1.5 Restarting or Deleting Instances in Batches.....	489
7.1.1.6 Querying All Instances.....	492
7.1.1.7 Creating a Topic in a Kafka Instance.....	498

7.1.1.8 Querying a Topic in a Kafka Instance.....	500
7.1.1.9 Deleting Topics in a Kafka Instance in Batches.....	503
7.1.2 Other APIs.....	504
7.1.2.1 Querying AZ Information.....	504
7.1.2.2 Querying Product Specifications.....	506
7.1.2.3 Querying Maintenance Time Windows.....	511
7.2 API V2.....	513
7.2.1 Managing Lifecycle.....	513
7.2.1.1 Querying an Instance.....	513
7.2.2 Instance Management.....	523
7.2.2.1 Adding Partitions to a Topic for a Kafka Instance.....	523
7.2.3 Managing Consumer Groups.....	524
7.2.3.1 Resetting Consumer Group Offset to the Specified Position.....	524
7.2.3.2 Querying All Consumer Groups.....	531
7.2.4 Smart Connect.....	534
7.2.4.1 Modifying Dumping Task Quotas.....	534
7.2.4.2 Creating an Order to Disable Dumping Connectors for an Instance.....	538
7.2.4.3 Creating a Dumping Task.....	542
7.2.4.4 Listing Dumping Tasks.....	549
7.2.4.5 Querying a Dumping Task.....	553
7.2.4.6 Deleting a Dumping Task.....	559
8 Appendix.....	563
8.1 Status Code.....	563
8.2 Error Codes.....	566
8.3 Instance Status.....	589
8.4 Obtaining a Project ID.....	590
8.5 Obtaining the Account Name and Account ID.....	591
A Change History.....	593

1 Before You Start

Welcome to *Distributed Message Service for Kafka API Reference*. Distributed Message Service (DMS) for Kafka is a message queuing service that is based on the open-source Apache Kafka. It provides Kafka premium instances with isolated computing, storage, and bandwidth resources. DMS for Kafka allows you to apply resources, configure topics, partitions, and replicas as required. The service can be used out of the box and frees you from deployment and O&M so that you can focus on the agile development of your applications.

This document describes the functions, syntax, parameters, and examples of the application programming interfaces (APIs) of DMS for Kafka.

DMS for Kafka supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Calling APIs](#).

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](#).

Concepts

- Account
An account is created upon successful registration with the cloud system. 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, which should not be used directly to perform routine management. To ensure account security, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- IAM user
An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).
The account name, username, and password will be required for API authentication.
- Region

Regions are divided from the dimensions of geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified into universal and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

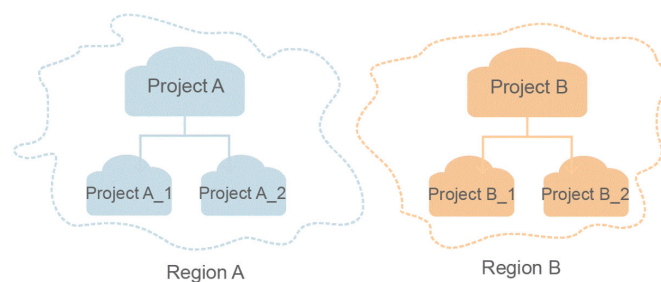
- Availability Zone (AZ)

An availability zone (AZ) comprises one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Compute, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

- Project

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

Figure 1-1 Project isolating model



- Enterprise project

Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple projects, and resources can be added to or removed from enterprise projects.

For details about how to obtain enterprise project IDs and features, see the [Enterprise Management User Guide](#).

2 API Overview

Table 2-1 APIs for managing instances

API	Description
Managing Lifecycle	Includes: <ul style="list-style-type: none">• Creating an Instance• Listing All Instances• Querying an Instance• Deleting an Instance• Modifying Instance Information• Batch Restarting or Deleting Instances• Obtaining Instance Configurations• Modifying Instance Configurations• Upgrading an Instance

API	Description
<p>Instance Management</p>	<p>Includes:</p> <ul style="list-style-type: none"> ● Resetting the Password ● Resetting Kafka Manager Password ● Restarting Kafka Manager ● Configuring Automatic Topic Creation ● Modifying the Private IP Address for Cross-VPC Access ● Querying Kafka Cluster Metadata ● Reassigning Replicas of a Topic for a Kafka Instance ● Reassigning Replicas of a Topic for a Kafka Instance ● Modifying Kafka Access Modes ● Querying the Disk Usage Status of Topics ● Initiating Partition Reassigning for a Kafka Instance ● Disabling Kafka Manager ● Deleting a User or Client Quota ● Querying User or Client Quotas ● Creating User or Client Quotas ● Modifying User or Client Quotas
<p>Smart Connect</p>	<p>Includes:</p> <ul style="list-style-type: none"> ● Enabling Smart Connect (Pay-per-Use Instance) ● Disabling Smart Connect (Pay-per-Use Instance) ● Creating a Smart Connect Task ● Querying Smart Connect Tasks ● Querying Smart Connect Task Details ● Deleting a Smart Connect Task ● Pausing a Smart Connect Task ● Restarting a Smart Connect Task ● Starting a Smart Connect Task or Restarting a Paused or Running Smart Connect Task
<p>Specification Modification Management</p>	<p>Includes:</p> <ul style="list-style-type: none"> ● Increasing Instance Specifications ● Querying Product Information for Instance Specification Modification

API	Description
Topic Management	Includes: <ul style="list-style-type: none"> ● Producing Messages to Kafka ● Creating a Topic for a Kafka Instance ● Listing Topics of a Kafka Instance ● Modifying Topics of a Kafka Instance ● Batch Deleting Topics of a Kafka Instance ● Querying the Partition List of a Topic ● Querying the Current Producer List of a Topic ● Querying Topic Details
Managing Consumer Groups	Includes: <ul style="list-style-type: none"> ● Querying Consumer Group Details ● Querying All Consumer Groups ● Deleting Consumer Groups of a Kafka Instance in Batches ● Creating a Consumer Group ● Resetting Consumer Group Offset to the Specified Position ● Querying the Offset of a Consumer Group ● Modifying All Consumer Groups ● Querying a Specified Consumer Group ● Deleting a Specified Consumer Group ● Modifying a Specified Consumer Group ● Querying Topics of a Specified Consumer Group ● Querying Consumers in a Specified Consumer Group
User Management	Includes: <ul style="list-style-type: none"> ● Querying the User List ● Creating a User ● Deleting Users in Batches ● Resetting a User Password ● Modifying User Parameters ● Querying User Permissions ● Granting User Permissions

API	Description
Managing Messages	Includes: <ul style="list-style-type: none"> ● Querying messages ● Querying a Message with a Specified Offset ● Querying a Message with a Specified Time Period ● Querying Offset of the Earliest Message in a Partition ● Querying Offset of the Latest Message in a Partition ● Deleting a Kafka Message
Background Task Management	Includes: <ul style="list-style-type: none"> ● Listing Background Tasks ● Querying a Background Task ● Deleting a Background Task
Tag Management	Includes: <ul style="list-style-type: none"> ● Batch Adding or Deleting Tags ● Listing Tags of an Instance ● Listing Tags of a Project
Diagnosis Management	Includes: <ul style="list-style-type: none"> ● Pre-check Before Diagnosing Message Stack ● Creating a Message Stack Diagnosis Task ● Querying the Message Stack Diagnosis Report List ● Batch Deleting Message Stack Diagnosis Reports ● Querying Diagnosis Report Details
Other APIs	Includes: <ul style="list-style-type: none"> ● Listing Maintenance Time Windows ● Listing AZ Information ● Querying Product Specifications List ● Querying Kafka Instance Monitoring Dimensions ● Querying vCPUs of a Kafka Flavor

3 Calling APIs

3.1 Making an API Request

This section describes how to make a REST API request, and uses the IAM API for [creating an IAM user](#) as an example to describe how to call an API.

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 Parameters in a URI

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in the CN-Hong Kong region is iam.ap-southeast-1.myhuaweicloud.com .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used for creating an IAM user is /v3.0/OS-USER/users .
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of <i>Parameter name=Parameter value</i> . For example, ?limit=10 indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint of IAM (**iam.ap-southeast-1.myhuaweicloud.com**) for this region and the resource-path (**/v3.0/OS-USER/users**) in the URI of the API for **creating an IAM user**. Then, construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

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:

- **GET:** requests a server to return specified resources.
- **PUT:** requests a server to update specified resources.
- **POST:** requests a server to add resources or perform special operations.
- **DELETE:** requests a server to delete specified resources, for example, objects.
- **HEAD:** same as GET except that the server must return only the response header.
- **PATCH:** requests a server to update a part of a specified resource. If the resource does not exist, a new resource can be created using the PATCH method.

In the URI of the API for **creating an IAM user**, the request method is **POST**, so the request is:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

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 3-2 lists the common request header fields.

Table 3-2 Common request header fields

Name	Description	Mandatory	Example Value
Host	Request server information, which is obtained from the URL of a service API. The value is in the format of <i>Hostname.Port number</i> . If no port is specified, the default port will be used. For HTTPS, port 443 is used by default.	No This parameter is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Message body type or format. The default value application/json is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Length of the request body. The unit is byte.	No	3495
X-Project-Id	Project ID. Obtain the project ID by following the instructions in Obtaining a Project ID .	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cb aa340f9c0f4

Name	Description	Mandatory	Example Value
X-Auth-Token	<p>User token.</p> <p>The user token is a response to the API used for obtaining a user token. This API is the only one that does not require authentication.</p> <p>After the request is processed, the value of X-Subject-Token in the response header is the token value.</p>	No Mandatory for token-based authentication.	The following is part of an example token: MIIPAgYJKoZlhvcNAQc-Co...ggg1BBIINPXsidG9rZ

 NOTE

In addition to supporting token-based authentication, cloud service APIs also support authentication using the AK/SK. During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see "AK/SK-based Authentication" in [Authentication](#).

The API used for **creating an IAM user** requires authentication using AK/SK. An example of such requests is as follows:

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

Request Body (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.

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

For an API for **creating an IAM user**, the required parameter and description are in the request part of the API. An example of such requests is as follows. Use actual values in the bold fields.

- **accountid** indicates the account ID of the IAM user.
- **username** indicates the username of the IAM user to be created.
- **email** indicates the email of the IAM user.
- ********* indicates the password of the IAM user.

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
```



```
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

If all data required by a request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding.

3.2 Authentication

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

- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK-based authentication is recommended because it is more secure than token-based authentication.
- Token-based authentication: Requests are authenticated using a token.

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 request headers for authentication.

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

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

NOTICE

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

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to request headers to get permissions for calling the API. You can obtain a token by [calling an API](#).

A cloud service can be deployed as either a project-level service or global service.

- For a project-level service, you need to obtain a project-level token. When you call the API, set **auth.scope** in the request body to **project**.
- For a global service, you need to obtain a global token. When you call the API, set **auth.scope** in the request body to **domain**.

When calling the API used for [obtaining a user token](#), you must set **auth.scope** in the request body to **project**.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", //IAM username.
          "password": $ADMIN_PASS, //IAM password. For security, you are advised to store it in
            ciphertext in the configuration file or environment variable.
          "domain": {
            "name": "domainname" //Name of the account of the IAM user.
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxx" //Project name.
      }
    }
  }
}
```

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.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

3.3 Returned Values

Status Code

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

A status code is a group of digits ranging from 1xx to 5xx. It indicates the status of a response. For more information, see [Status Code](#).

If status code **201** is returned for the calling of the API for [obtaining a user token](#), the request is successful.

Response Header

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

[Figure 3-2](#) shows the response header for the API of [creating an IAM user](#).

Figure 3-2 Header of the response to the request for creating an IAM user

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fcec8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_██████████_null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

(Optional) Response Body

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

The response body for the API of [creating an IAM user](#) is shown as follows. The following shows part of the response body for the API to obtain a user token.

```
{
  "user": {
    "id": "c131886aec...",
    "name": "IAMUser",
    "description": "IAM User Description",
    "areacode": "",
    "phone": "",
    "email": "***@***.com",
    "status": null,
    "enabled": true,
    "pwd_status": false,
    "access_mode": "default",
    "is_domain_owner": false,
    "xuser_id": "",
    "xuser_type": "",
    "password_expires_at": null,
    "create_time": "2024-05-21T09:03:41.000000",
    "domain_id": "d78cbac1.....",
    "xdomain_id": "30086000.....",
    "xdomain_type": "",
    "default_project_id": null
  }
}
```

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

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

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

4 Getting Started

Scenarios

This section describes how to call an API to create a Kafka instance and customize the computing capabilities and storage space of the instance based on service requirements.

For details on how to call APIs, see [Calling APIs](#).

Prerequisites

- IAM endpoint obtained from [Regions and Endpoints](#).
- Kafka endpoint obtained from [Regions and Endpoints](#).

Creating a Kafka Instance

The following is an example request for creating a Kafka instance:

```
{
  "name": "kafka-demo",
  "engine": "kafka",
  "engine_version": 2.7,
  "broker_num": 3,
  "storage_space": 300,
  "vpc_id": "ead6c5ff-xxx-9ba91820e72c",
  "security_group_id": "aa75ae22-xxx-a9dec8c73220",
  "subnet_id": "3cb6afa2-xxx-05a7f671d6a8",
  "available_zones": [
    "effdbcxxx6b42f56533"
  ],
  "product_id": "c6.2u4g.cluster",
  "storage_spec_code": "dms.physical.storage.high.v2"
}
```

- **name**: name of the instance.
- **engine**: message engine. The value is **kafka**.
- **engine_version**: version of the message engine.
- **broker_num**: number of proxies.
- **storage_space**: message storage space in GB. For details about the value range, see [Creating an Instance](#).
- **vpc_id**: ID of the VPC where the Kafka instance resides. Obtain the value by using the API described in [Creating an Instance](#).

- **security_group_id**: ID of the security group. Obtain the value by using the API described in [Creating an Instance](#).
- **subnet_id**: ID of the VPC subnet. Obtain the value by using the API described in [Creating an Instance](#).
- **available_zones**: ID of the AZ where the instance resides. The value cannot be empty or null. Obtain the value by calling the API described in [Querying AZ Information](#).
- **product_id**: ID of the product. Obtain the value by calling the API described in [Querying Product Specifications](#).
- **storage_spec_code**: storage I/O specification. For details about the value range, see [Creating an Instance](#).

5 APIs V2 (Recommended)

5.1 Managing Lifecycle

5.1.1 Creating an Instance

Function

This API is used to create an instance.

Pay-per-use and yearly/monthly billing modes are available.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{engine}/{project_id}/instances

Table 5-1 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .

Request Parameters

Table 5-2 Request body parameters

Parameter	Mandatory	Type	Description
name	Yes	String	Instance name. An instance name can contain 4 to 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
description	No	String	Description of an instance. The description supports up to 1024 characters. NOTE The backslash (\) and quotation mark (") are special characters for JSON messages. When using these characters in a parameter value, add the escape character (\) before the characters, for example, \ and \".
engine	Yes	String	Message engine. Value: kafka .
engine_version	Yes	String	Message engine. Values: <ul style="list-style-type: none"> • 1.1.0 • 2.7 • 3.x
broker_num	Yes	Integer	Number of brokers.
storage_space	Yes	Integer	Message storage space, in GB. <ul style="list-style-type: none"> • 300–300,000 GB for c6.2u4g.cluster Kafka instances • 300–600,000 GB for c6.4u8g.cluster Kafka instances • 300–1,500,000 GB for c6.8u16g.cluster Kafka instances • 300–1,500,000 GB for c6.12u24g.cluster Kafka instances • 300–1,500,000 GB for c6.16u32g.cluster Kafka instances

Parameter	Mandatory	Type	Description
access_user	No	String	<p>This parameter is mandatory when ssl_enable is set to true. This parameter is invalid when ssl_enable is set to false.</p> <p>A username must start with a letter and only letters, digits, hyphens (-), and underscores (_) are allowed. It can contain 4 to 64 characters.</p>
password	No	String	<p>This parameter is mandatory when ssl_enable is set to true. This parameter is invalid when ssl_enable is set to false.</p> <p>Instance password.</p> <p>The password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Can contain 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters include (~!@#\$%^&*()-_+ [{}]:'";<.>/?) <p>and spaces, and cannot start with a hyphen (-).</p>
vpc_id	Yes	String	<p>VPC ID.</p> <p>To obtain it, log in to the VPC console and view the VPC ID on the VPC details page.</p>
security_group_id	Yes	String	<p>ID of the security group to which the instance belongs.</p> <p>To obtain it, log in to the VPC console and view the security group ID on the security group details page.</p>

Parameter	Mandatory	Type	Description
subnet_id	Yes	String	Subnet information. To obtain it, log in to VPC console and click the target subnet on the Subnets page. You can view the network ID on the displayed page.
available_zones	Yes	Array of strings	ID of the AZ where instance brokers reside and which has available resources. Obtain the AZ ID by referring to Listing AZ Information . This parameter cannot be empty or null. A Kafka instance can be deployed in 1 AZ or at least 3 AZs. When specifying AZs for brokers, use commas (,) to separate multiple AZs.
product_id	Yes	String	Product ID. Obtain the product ID from Querying Product Specifications List .
maintain_begin	No	String	Time at which the maintenance window starts. The format is HH:mm.
maintain_end	No	String	Time at which the maintenance window ends. The format is HH:mm.
enable_publicip	No	Boolean	Whether to enable public access. By default, public access is disabled. <ul style="list-style-type: none"> ● true: enable ● false: disable

Parameter	Mandatory	Type	Description
tenant_ips	No	Array of strings	<p>You can manually specify IPv4 private IP addresses when creating an instance.</p> <p>The number of specified private IP addresses must be equal to or less than the number of created brokers.</p> <p>If the number of specified private IP addresses is less than the number of created brokers, the system automatically assigns private IP addresses to the remaining brokers.</p>
publicip_id	No	String	<p>ID of the EIP bound to the instance.</p> <p>Use commas (,) to separate multiple EIP IDs.</p> <p>This parameter is mandatory if public access is enabled (that is, enable_publicip is set to true).</p>
ssl_enable	No	Boolean	<p>Whether SASL is enabled.</p> <ul style="list-style-type: none">● true: enable● false: disable

Parameter	Mandatory	Type	Description
kafka_security_protocol	No	String	<p>Security protocol to use after SASL is enabled.</p> <ul style="list-style-type: none"> • SASL_SSL: Data is encrypted with SSL certificates for high-security transmission. • SASL_PLAINTEXT: Data is transmitted in plaintext with username and password authentication. <p>If this parameter is left blank, SASL_SSL authentication is enabled by default. This parameter cannot be manually modified once the instance is created.</p> <p>If the port_protocol parameter were used in instance creation, its value is used for the private and public network access security protocols and this parameter becomes invalid.</p>
sasl_enabled_mechanisms	No	Array of strings	<p>Authentication mechanism to use after SASL is enabled. This parameter is mandatory if SASL authentication is enabled (ssl_enable=true).</p> <p>If this parameter is left blank, PLAIN authentication is enabled by default.</p> <p>Select both or either of the following mechanisms for SASL authentication.</p> <p>Options:</p> <ul style="list-style-type: none"> • PLAIN: simple username and password verification. • SCRAM-SHA-512: user credential verification, which is more secure than PLAIN.

Parameter	Mandatory	Type	Description
port_protocol	No	PortProtocol object	Kafka instance access mode. PLAINTEXT indicates plaintext access. SASL_SSL or SASL_PLAINTEXT indicates ciphertext access. Once enabled, this parameter cannot be disabled. Enable plaintext or ciphertext access, or both. The security protocol for cross-VPC access is the same as that for intranet access. If both ciphertext access and plaintext access are enabled for intranet access, the security protocol for ciphertext access is preferentially used for cross-VPC access.
retention_policy	No	String	Action to be taken when the memory usage reaches the disk capacity threshold. The value can be: <ul style="list-style-type: none"> • produce_reject: New messages cannot be created. • time_base: The earliest messages are deleted.
ipv6_enable	No	Boolean	Indicates whether to enable IPv6. This parameter is available only when the VPC supports IPv6.
disk_encrypted_enable	No	Boolean	Indicates whether to enable disk encryption.
disk_encrypted_key	No	String	Disk encryption key. If disk encryption is not enabled, this parameter is left blank.
connector_enable	No	Boolean	Whether to enable message dumping. By default, message dumping is disabled.

Parameter	Mandatory	Type	Description
enable_auto_topic	No	Boolean	Whether to enable automatic topic creation. <ul style="list-style-type: none"> • true: enable • false: disable If it is enabled, a topic will be automatically created with 3 partitions and 3 replicas when a message is created in or retrieved from a topic that does not exist. The default value is false .
storage_spec_code	Yes	String	Storage I/O specification. Value range: <ul style="list-style-type: none"> • dms.physical.storage.high.v2: high I/O • dms.physical.storage.ultra.v2: ultra-high I/O For details about how to select a disk type, see "Disk Types and Disk Performance" in the EVS Service Overview.
enterprise_project_id	No	String	Enterprise project ID. This parameter is mandatory for an enterprise project account.
tags	No	Array of TagEntity objects	Tag list.
arch_type	No	String	CPU architecture. Currently supported: x86. Value range: <ul style="list-style-type: none"> • X86
vpc_client_plain	No	Boolean	Intra-VPC plaintext access.
bss_param	No	BssParam object	Parameter related to the yearly/monthly billing mode. If this parameter is left blank, the billing mode is pay-per-use by default. If this parameter is not left blank, the billing mode is yearly/monthly.

Table 5-3 PortProtocol

Parameter	Mandatory	Type	Description
private_plain_enable	No	Boolean	<p>Whether to enable the private plaintext access mode.</p> <p>Options:</p> <ul style="list-style-type: none"> • true: Enabled. The connection address is <i>ip:9092</i> and the access protocol is PLAINTEXT. • false: Disabled. <p>The default option is false.</p>
private_sasl_ssl_enable	No	Boolean	<p>Whether to enable the private ciphertext access mode using the security protocol SASL_SSL.</p> <p>Options:</p> <ul style="list-style-type: none"> • true: Enabled. private_sasl_ssl_enable and private_sasl_plaintext_enable cannot be set to true at the same time. • false: Disabled. <p>The default option is false.</p>
private_sasl_plaintext_enable	No	Boolean	<p>Whether to enable the private ciphertext access mode using the security protocol SASL_PLAINTEXT.</p> <p>Options:</p> <ul style="list-style-type: none"> • true: Enabled. The connection address is <i>ip:9093</i> and the access protocol is SASL_PLAINTEXT. private_sasl_plaintext_enable and private_sasl_ssl_enable cannot be set to true at the same time. • false: Disabled. <p>The default option is false.</p>

Parameter	Mandatory	Type	Description
public_plain_enable	No	Boolean	<p>Whether to enable the public plaintext access mode.</p> <p>Options:</p> <ul style="list-style-type: none"> • true: Enabled. The connection address is <i>ip:9094</i> and the access protocol is PLAINTEXT. Enable public access before enabling public plaintext access. • false: Disabled. <p>The default option is false.</p>
public_sasl_ssl_enable	No	Boolean	<p>Whether to enable the public ciphertext access using the security protocol SASL_SSL.</p> <p>Options:</p> <ul style="list-style-type: none"> • true: Enabled. The connection address is <i>ip:9095</i> and the access protocol is SASL_SSL. public_sasl_ssl_enable and public_sasl_plaintext_enable cannot be set to true at the same time. If this parameter is set to true, public access needs to be enabled for the instance. • false: Disabled. <p>The default option is false.</p>

Parameter	Mandatory	Type	Description
public_sasl_plaintext_enable	No	Boolean	<p>Whether to enable the public ciphertext access mode using the security protocol SASL_PLAINTEXT.</p> <p>Value range:</p> <ul style="list-style-type: none"> true: Enabled. The connection address is <i>ip:9095</i> and the access protocol is SASL_PLAINTEXT. <p>public_sasl_plaintext_enable and public_sasl_ssl_enable cannot be set to true at the same time.</p> <p>If this parameter is set to true, public access needs to be enabled for the instance.</p> <ul style="list-style-type: none"> false: Disabled. <p>The default value is false.</p>

Table 5-4 TagEntity

Parameter	Mandatory	Type	Description
key	No	String	<p>Tag key.</p> <ul style="list-style-type: none"> Cannot be left blank. Must be unique for the same instance. Can contain 1 to 128 characters. Can contain letters, digits, spaces, and special characters <code>._:=-+@</code> Cannot start with sys Cannot start or end with a space.
value	No	String	<p>Tag value.</p> <ul style="list-style-type: none"> Can contain 0 to 255 characters. Can contain letters, digits, spaces, and special characters <code>._:=-+@</code>

Table 5-5 BssParam

Parameter	Mandatory	Type	Description
is_auto_renew	No	Boolean	Whether auto renewal is enabled. Options: <ul style="list-style-type: none"> • true: Auto renewal is enabled. • false: Auto renewal is not enabled. By default, auto renewal is disabled.
charging_mode	No	String	Billing mode. This parameter specifies a payment mode. Options: <ul style="list-style-type: none"> • prePaid: yearly/monthly billing. • postPaid: pay-per-use billing. The default value is postPaid .
is_auto_pay	No	Boolean	Specifies whether the order is automatically or manually paid. Options: <ul style="list-style-type: none"> • true: The order will be automatically paid. • false: The order must be manually paid. The default payment mode is manual.
period_type	No	String	Subscription period type. Options: <ul style="list-style-type: none"> • month • year This parameter is valid and mandatory only when chargingMode is set to prePaid . **

Parameter	Mandatory	Type	Description
period_num	No	Integer	Subscribed periods. Options: <ul style="list-style-type: none"> If periodType is month, the value ranges from 1 to 9. If periodType is year, the value ranges from 1 to 3. **This parameter is valid and mandatory only when chargingMode is set to prePaid.**

Response Parameters

Status code: 200

Table 5-6 Response body parameters

Parameter	Type	Description
instance_id	String	Instance ID.

Example Requests

- Creating a pay-per-use Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

POST https://{endpoint}/v2/{engine}/{project_id}/instances

```
{
  "name": "kafka-test",
  "description": "",
  "engine": "kafka",
  "engine_version": "2.7",
  "storage_space": 300,
  "vpc_id": "*****-9b4a-44c5-a964-*****",
  "subnet_id": "*****-8fbf-4438-ba71-*****",
  "security_group_id": "*****-e073-4aad-991f-*****",
  "available_zones": [ "*****706d4c1fb0eb72f0*****" ],
  "product_id": "c6.2u4g.cluster",
  "ssl_enable": true,
  "kafka_security_protocol": "SASL_SSL",
  "sas_enabled_mechanisms": [ "SCRAM-SHA-512" ],
  "storage_spec_code": "dms.physical.storage.ultra.v2",
  "broker_num": 3,
  "arch_type": "X86",
  "enterprise_project_id": "0",
  "access_user": "*****",
  "password": "*****",
  "enable_publicip": true,
  "tags": [ {
    "key": "aaa",
    "value": "111"
  } ],
}
```

```
"retention_policy" : "time_base",
"disk_encrypted_enable" : true,
"disk_encrypted_key" : "*****-b953-4875-a743-*****",
"publicip_id" : "*****-88fc-4a8c-86d0-*****, *****-16af-455d-8d54-
*****_3d69-4367-95ab-*****",
"vpc_client_plain" : true,
"enable_auto_topic" : true,
"tenant_ips" : [ "127.xx.xx.x", "127.xx.xx.x", "127.xx.xx.x" ]
}
```

- Creating a yearly/monthly Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

POST https://{endpoint}/v2/{engine}/{project_id}/instances

```
{
"name": "kafka-test1",
"description": "",
"engine": "kafka",
"engine_version": "2.7",
"storage_space": 300,
"vpc_id": "*****-9b4a-44c5-a964-*****",
"subnet_id": "*****-8fbf-4438-ba71-*****",
"security_group_id": "*****-e073-4aad-991f-*****",
"available_zones": [ "*****706d4c1fb0eb72f0*****" ],
"product_id": "c6.2u4g.cluster",
"ssl_enable": true,
"kafka_security_protocol": "SASL_SSL",
"sasl_enabled_mechanisms": [ "SCRAM-SHA-512" ],
"storage_spec_code": "dms.physical.storage.ultra.v2",
"broker_num": 3,
"arch_type": "X86",
"enterprise_project_id": "0",
"access_user": "*****",
"password": "*****",
"enable_publicip": true,
"tags": [ {
"key": "aaa",
"value": "111"
} ],
"retention_policy": "time_base",
"publicip_id": "*****-88fc-4a8c-86d0-*****, *****-16af-455d-8d54-
*****_3d69-4367-95ab-*****",
"vpc_client_plain": true,
"enable_auto_topic": true,
"bss_param": {
"charging_mode": "prePaid",
"period_type": "month",
"period_num": 1,
"is_auto_pay": true
},
"tenant_ips": [ "127.xx.xx.x", "127.xx.xx.x", "127.xx.xx.x" ]
}
```

Example Responses

Status code: 200

Instance created successfully.

```
{
"instance_id": "8959ab1c-7n1a-yyb1-a05t-93dfc361b32d"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Creating a pay-per-use Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceByEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateInstanceByEngineRequest request = new CreateInstanceByEngineRequest();
        request.withEngine(CreateInstanceByEngineRequest.EngineEnum.fromValue("{engine}"));
        CreateInstanceByEngineReq body = new CreateInstanceByEngineReq();
        List<TagEntity> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagEntity()
                .withKey("aaa")
                .withValue("111")
        );
        List<CreateInstanceByEngineReq.SaslEnabledMechanismsEnum> listbodySaslEnabledMechanisms
        = new ArrayList<>();

        listbodySaslEnabledMechanisms.add(CreateInstanceByEngineReq.SaslEnabledMechanismsEnum.fromV
        alue("SCRAM-SHA-512"));
        List<String> listbodyAvailableZones = new ArrayList<>();
        listbodyAvailableZones.add("*****706d4c1fb0eb72f0*****");
        body.withVpcClientPlain(true);
        body.withArchType("X86");
        body.withTags(listbodyTags);
        body.withEnterpriseProjectId("0");

        body.withStorageSpecCode(CreateInstanceByEngineReq.StorageSpecCodeEnum.fromValue("dms.physic
        al.storage.ultra.v2"));
        body.withEnableAutoTopic(true);
        body.withDiskEncryptedKey("*****.b953-4875-a743-*****");
        body.withDiskEncryptedEnable(true);

        body.withRetentionPolicy(CreateInstanceByEngineReq.RetentionPolicyEnum.fromValue("time_base"));
        body.withSaslEnabledMechanisms(listbodySaslEnabledMechanisms);
```

```
body.withKafkaSecurityProtocol("SASL_SSL");
body.withSslEnable(true);
body.withPublicId("*****-88fc-4a8c-86d0-*****;*****-16af-455d-8d54-
*****;*****_3d69-4367-95ab-*****");
body.withEnablePublicip(true);
body.withProductId("c6.2u4g.cluster");
body.withAvailableZones(listbodyAvailableZones);
body.withSubnetId("*****-8fbf-4438-ba71-*****");
body.withSecurityGroupId("*****-e073-4aad-991f-*****");
body.withVpcId("*****-9b4a-44c5-a964-*****");
body.withPassword("*****");
body.withAccessUser("*****");
body.withStorageSpace(300);
body.withBrokerNum(3);
body.withEngineVersion("2.7");
body.withEngine(CreateInstanceByEngineReq.EngineEnum.fromValue("kafka"));
body.withDescription("");
body.withName("kafka-test");
request.withBody(body);
try {
    CreateInstanceByEngineResponse response = client.createInstanceByEngine(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Creating a yearly/monthly Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceByEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateInstanceByEngineRequest request = new CreateInstanceByEngineRequest();
request.withEngine(CreateInstanceByEngineRequest.EngineEnum.fromValue("{engine}"));
CreateInstanceByEngineReq body = new CreateInstanceByEngineReq();
BssParam bssParambody = new BssParam();
bssParambody.withChargingMode(BssParam.ChargingModeEnum.fromValue("prePaid"))
    .withIsAutoPay(true)
    .withPeriodType(BssParam.PeriodTypeEnum.fromValue("month"))
    .withPeriodNum(1);
List<TagEntity> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new TagEntity()
        .withKey("aaa")
        .withValue("111")
);
List<CreateInstanceByEngineReq.SaslEnabledMechanismsEnum> listbodySaslEnabledMechanisms
= new ArrayList<>();

listbodySaslEnabledMechanisms.add(CreateInstanceByEngineReq.SaslEnabledMechanismsEnum.fromV
alue("SCRAM-SHA-512"));
List<String> listbodyAvailableZones = new ArrayList<>();
listbodyAvailableZones.add("*****706d4c1fb0eb72f0*****");
body.withBssParam(bssParambody);
body.withVpcClientPlain(true);
body.withArchType("X86");
body.withTags(listbodyTags);
body.withEnterpriseProjectId("0");

body.withStorageSpecCode(CreateInstanceByEngineReq.StorageSpecCodeEnum.fromValue("dms.physic
al.storage.ultra.v2"));
body.withEnableAutoTopic(true);

body.withRetentionPolicy(CreateInstanceByEngineReq.RetentionPolicyEnum.fromValue("time_base"));
body.withSaslEnabledMechanisms(listbodySaslEnabledMechanisms);
body.withKafkaSecurityProtocol("SASL_SSL");
body.withSslEnable(true);
body.withPublicIpId("*****_88fc-4a8c-86d0-*****; *****-16af-455d-8d54-
*****_3d69-4367-95ab-*****");
body.withEnablePublicIp(true);
body.withProductId("c6.2u4g.cluster");
body.withAvailableZones(listbodyAvailableZones);
body.withSubnetId("*****-8fbf-4438-ba71-*****");
body.withSecurityGroupId("*****_e073-4aad-991f-*****");
body.withVpcId("*****_9b4a-44c5-a964-*****");
body.withPassword("*****");
body.withAccessUser("*****");
body.withStorageSpace(300);
body.withBrokerNum(3);
body.withEngineVersion("2.7");
body.withEngine(CreateInstanceByEngineReq.EngineEnum.fromValue("kafka"));
body.withDescription("");
body.withName("kafka-test1");
request.withBody(body);
try {
    CreateInstanceByEngineResponse response = client.createInstanceByEngine(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

- Creating a pay-per-use Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceByEngineRequest()
        request.engine = "{engine}"
        listTagsbody = [
            TagEntity(
                key="aaa",
                value="111"
            )
        ]
        listSaslEnabledMechanismsbody = [
            "SCRAM-SHA-512"
        ]
        listAvailableZonesbody = [
            "*****706d4c1fb0eb72f0*****"
        ]
        request.body = CreateInstanceByEngineReq(
            vpc_client_plain=True,
            arch_type="X86",
            tags=listTagsbody,
            enterprise_project_id="0",
            storage_spec_code="dms.physical.storage.ultra.v2",
            enable_auto_topic=True,
            disk_encrypted_key="*****-b953-4875-a743-*****",
            disk_encrypted_enable=True,
            retention_policy="time_base",
            sasl_enabled_mechanisms=listSaslEnabledMechanismsbody,
            kafka_security_protocol="SASL_SSL",
            ssl_enable=True,
            publicip_id="*****-88fc-4a8c-86d0-*****, *****-16af-455d-8d54-
            *****-3d69-4367-95ab-*****",
            enable_publicip=True,
            product_id="c6.2u4g.cluster",
            available_zones=listAvailableZonesbody,
```



```
        subnet_id="*****-8fbf-4438-ba71-*****",
        security_group_id="*****-e073-4aad-991f-*****",
        vpc_id="*****-9b4a-44c5-a964-*****",
        password="*****",
        access_user="*****",
        storage_space=300,
        broker_num=3,
        engine_version="2.7",
        engine="kafka",
        description="",
        name="kafka-test"
    )
    response = client.create_instance_by_engine(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Creating a yearly/monthly Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceByEngineRequest()
        request.engine = "{engine}"
        bssParambody = BssParam(
            charging_mode="prePaid",
            is_auto_pay=True,
            period_type="month",
            period_num=1
        )
        listTagsbody = [
            TagEntity(
                key="aaa",
                value="111"
            )
        ]
        listSaslEnabledMechanismsbody = [
            "SCRAM-SHA-512"
        ]
        listAvailableZonesbody = [
            "*****706d4c1fb0eb72f0*****"
        ]
    ]
```

```
request.body = CreateInstanceByEngineReq(  
    bss_param=bssParambody,  
    vpc_client_plain=True,  
    arch_type="X86",  
    tags=listTagsbody,  
    enterprise_project_id="0",  
    storage_spec_code="dms.physical.storage.ultra.v2",  
    enable_auto_topic=True,  
    retention_policy="time_base",  
    sasl_enabled_mechanisms=listSaslEnabledMechanismsbody,  
    kafka_security_protocol="SASL_SSL",  
    ssl_enable=True,  
    publicip_id="*****-88fc-4a8c-86d0-*****;*****-16af-455d-8d54-  
*****-3d69-4367-95ab-*****",  
    enable_publicip=True,  
    product_id="c6.2u4g.cluster",  
    available_zones=listAvailableZonesbody,  
    subnet_id="*****-8fbf-4438-ba71-*****",  
    security_group_id="*****-e073-4aad-991f-*****",  
    vpc_id="*****-9b4a-44c5-a964-*****",  
    password="*****",  
    access_user="*****",  
    storage_space=300,  
    broker_num=3,  
    engine_version="2.7",  
    engine="kafka",  
    description="",  
    name="kafka-test1"  
)  
response = client.create_instance_by_engine(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

- Creating a pay-per-use Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
    environment variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before  
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
    environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateInstanceByEngineRequest{}
request.Engine = model.GetCreateInstanceByEngineRequestEngineEnum().ENGINE
keyTags:= "aaa"
valueTags:= "111"
var listTagsbody = []model.TagEntity{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
}
var listSaslEnabledMechanismsbody = []model.CreateInstanceByEngineReqSaslEnabledMechanisms{
    model.GetCreateInstanceByEngineReqSaslEnabledMechanismsEnum().SCRAM_SHA_512,
}
var listAvailableZonesbody = []string{
    "*****706d4c1fb0eb72f0*****",
}
vpcClientPlainCreateInstanceByEngineReq:= true
archTypeCreateInstanceByEngineReq:= "X86"
enterpriseProjectIdCreateInstanceByEngineReq:= "0"
enableAutoTopicCreateInstanceByEngineReq:= true
diskEncryptedKeyCreateInstanceByEngineReq:= "*****b953-4875-a743-*****"
diskEncryptedEnableCreateInstanceByEngineReq:= true
retentionPolicyCreateInstanceByEngineReq:=
model.GetCreateInstanceByEngineReqRetentionPolicyEnum().TIME_BASE
kafkaSecurityProtocolCreateInstanceByEngineReq:= "SASL_SSL"
sslEnableCreateInstanceByEngineReq:= true
publicIpIdCreateInstanceByEngineReq:= "*****-88fc-4a8c-86d0-*****", "*****-16af-455d-8d54-
*****_3d69-4367-95ab-*****"
enablePublicIpCreateInstanceByEngineReq:= true
passwordCreateInstanceByEngineReq:= "*****"
accessUserCreateInstanceByEngineReq:= "*****"
descriptionCreateInstanceByEngineReq:= ""
request.Body = &model.CreateInstanceByEngineReq{
    VpcClientPlain: &vpcClientPlainCreateInstanceByEngineReq,
    ArchType: &archTypeCreateInstanceByEngineReq,
    Tags: &listTagsbody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceByEngineReq,
    StorageSpecCode:
model.GetCreateInstanceByEngineReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_ULTRA,
    EnableAutoTopic: &enableAutoTopicCreateInstanceByEngineReq,
    DiskEncryptedKey: &diskEncryptedKeyCreateInstanceByEngineReq,
    DiskEncryptedEnable: &diskEncryptedEnableCreateInstanceByEngineReq,
    RetentionPolicy: &retentionPolicyCreateInstanceByEngineReq,
    SaslEnabledMechanisms: &listSaslEnabledMechanismsbody,
    KafkaSecurityProtocol: &kafkaSecurityProtocolCreateInstanceByEngineReq,
    SslEnable: &sslEnableCreateInstanceByEngineReq,
    PublicIpId: &publicIpIdCreateInstanceByEngineReq,
    EnablePublicIp: &enablePublicIpCreateInstanceByEngineReq,
    ProductId: "c6.2u4g.cluster",
    AvailableZones: listAvailableZonesbody,
    SubnetId: "*****-8fbf-4438-ba71-*****",
    SecurityGroupId: "*****-e073-4aad-991f-*****",
    Vpclid: "*****-9b4a-44c5-a964-*****",
    Password: &passwordCreateInstanceByEngineReq,
    AccessUser: &accessUserCreateInstanceByEngineReq,
    StorageSpace: int32(300),
    BrokerNum: int32(3),
    EngineVersion: "2.7",
    Engine: model.GetCreateInstanceByEngineReqEngineEnum().KAFKA,
    Description: &descriptionCreateInstanceByEngineReq,
    Name: "kafka-test",
}
response, err := client.CreateInstanceByEngine(request)

```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Creating a yearly/monthly Kafka instance whose version is 2.7, specifications are 2 vCPUs | 4 GB x 3, and storage space is 300 GB

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceByEngineRequest{}
    request.Engine = model.GetCreateInstanceByEngineRequestEngineEnum().ENGINE
    chargingModeBssParam := model.GetBssParamChargingModeEnum().PRE_PAID
    isAutoPayBssParam := true
    periodTypeBssParam := model.GetBssParamPeriodTypeEnum().MONTH
    periodNumBssParam := int32(1)
    bssParambody := &model.BssParam{
        ChargingMode: &chargingModeBssParam,
        IsAutoPay: &isAutoPayBssParam,
        PeriodType: &periodTypeBssParam,
        PeriodNum: &periodNumBssParam,
    }
    keyTags := "aaa"
    valueTags := "111"
    var listTagsbody = []model.TagEntity{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    var listSaslEnabledMechanismsbody = []model.CreateInstanceByEngineReqSaslEnabledMechanisms{
        model.GetCreateInstanceByEngineReqSaslEnabledMechanismsEnum().SCRAM_SHA_512,
    }
    var listAvailableZonesbody = []string{
        "*****706d4c1fb0eb72f0*****",
    }
    vpcClientPlainCreateInstanceByEngineReq := true
```

```

archTypeCreateInstanceByEngineReq:= "X86"
enterpriseProjectIdCreateInstanceByEngineReq:= "0"
enableAutoTopicCreateInstanceByEngineReq:= true
retentionPolicyCreateInstanceByEngineReq:=
model.GetCreateInstanceByEngineReqRetentionPolicyEnum().TIME_BASE
kafkaSecurityProtocolCreateInstanceByEngineReq:= "SASL_SSL"
sslEnableCreateInstanceByEngineReq:= true
publicIdCreateInstanceByEngineReq:= "*****-88fc-4a8c-86d0-***** , *****-16af-455d-8d54-
*****_3d69-4367-95ab_*****"
enablePublicIpCreateInstanceByEngineReq:= true
passwordCreateInstanceByEngineReq:= "*****"
accessUserCreateInstanceByEngineReq:= "*****"
descriptionCreateInstanceByEngineReq:= ""
request.Body = &model.CreateInstanceByEngineReq{
    BssParam: bssParambody,
    VpcClientPlain: &vpcClientPlainCreateInstanceByEngineReq,
    ArchType: &archTypeCreateInstanceByEngineReq,
    Tags: &listTagsbody,
    EnterpriseProjectId: &enterpriseProjectIdCreateInstanceByEngineReq,
    StorageSpecCode:
model.GetCreateInstanceByEngineReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_ULTRA,
    EnableAutoTopic: &enableAutoTopicCreateInstanceByEngineReq,
    RetentionPolicy: &retentionPolicyCreateInstanceByEngineReq,
    SaslEnabledMechanisms: &listSaslEnabledMechanismsbody,
    KafkaSecurityProtocol: &kafkaSecurityProtocolCreateInstanceByEngineReq,
    SslEnable: &sslEnableCreateInstanceByEngineReq,
    PublicId: &publicIdCreateInstanceByEngineReq,
    EnablePublicIp: &enablePublicIpCreateInstanceByEngineReq,
    ProductId: "c6.2u4g.cluster",
    AvailableZones: listAvailableZonesbody,
    SubnetId: "*****-8fbf-4438-ba71-*****",
    SecurityGroupId: "*****-e073-4aad-991f-*****",
    VpcId: "*****-9b4a-44c5-a964-*****",
    Password: &passwordCreateInstanceByEngineReq,
    AccessUser: &accessUserCreateInstanceByEngineReq,
    StorageSpace: int32(300),
    BrokerNum: int32(3),
    EngineVersion: "2.7",
    Engine: model.GetCreateInstanceByEngineReqEngineEnum().KAFKA,
    Description: &descriptionCreateInstanceByEngineReq,
    Name: "kafka-test1",
}
response, err := client.CreateInstanceByEngine(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Instance created successfully.

Error Codes

See [Error Codes](#).

5.1.2 Listing All Instances

Function

This API is used to query the instances of an account by the specified conditions.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances

Table 5-7 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .

Table 5-8 Query Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine. Value: kafka .
name	No	String	Instance name.
instance_id	No	String	Instance ID.
status	No	String	Instance status. For details, see Instance Status .
include_failure	No	String	Indicates whether to return the number of instances that fail to be created. If the value is true , the number of instances that failed to be created is returned. If the value is false , the number is not returned.

Parameter	Mandatory	Type	Description
exact_match_name	No	String	Whether to search for the instance that precisely matches a specified instance name. The default value is <i>false</i> , indicating that a fuzzy search is performed based on a specified instance name. If the value is true , the instance that precisely matches a specified instance name is queried.
enterprise_project_id	No	String	Enterprise project ID.
offset	No	String	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	String	Maximum number of instances returned in the current query. The default value is 10 . The value ranges from 1 to 50.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-9 Response body parameters

Parameter	Type	Description
instances	Array of ShowInstanceResp objects	Instance list.
instance_num	Integer	Number of instances.

Table 5-10 ShowInstanceResp

Parameter	Type	Description
name	String	Instance name.
engine	String	Message engine.
engine_version	String	Version.
description	String	Instance description
specification	String	Instance specifications.
storage_space	Integer	Message storage space in GB.
partition_num	String	Number of partitions in a Kafka instance.
used_storage_space	Integer	Used message storage space in GB.
dns_enable	Boolean	Indicates whether domain name access is enabled for an instance. <ul style="list-style-type: none"> ● true: enable ● false: disable
connect_address	String	IP address of an instance.
port	Integer	Port of an instance.
status	String	Instance status. For details, see Instance Status .
instance_id	String	Instance ID.
resource_spec_code	String	Resource specifications. <ul style="list-style-type: none"> ● dms.instance.kafka.cluster.c3.mini: Kafka instance with 100 MB/s bandwidth ● dms.instance.kafka.cluster.c3.small.2: Kafka instance with 300 MB/s bandwidth ● dms.instance.kafka.cluster.c3.middle.2: Kafka instance with 600 MB/s bandwidth ● dms.instance.kafka.cluster.c3.high.2: Kafka instance with 1200 MB/s bandwidth
charging_mode	Integer	Billing mode. Options: 1 : pay-per-use; 0 : yearly/monthly.
vpc_id	String	VPC ID.
vpc_name	String	VPC name.

Parameter	Type	Description
created_at	String	Time when the instance was created. The time is in the format of timestamp, that is, the offset milliseconds from 1970-01-01 00:00:00 UTC to the specified time.
subnet_name	String	Subnet name.
subnet_cidr	String	Subnet CIDR block.
user_id	String	User ID.
user_name	String	Username.
access_user	String	Username for accessing the instance.
order_id	String	Order ID. This parameter has a value only when the billing mode is yearly/monthly.
maintain_begin	String	Time at which the maintenance time window starts. The format is HH:mm:ss.
maintain_end	String	Time at which the maintenance time window ends. The format is HH:mm:ss.
enable_publicip	Boolean	Whether public access is enabled for the instance. <ul style="list-style-type: none"> • true: enabled • false: disabled
management_connect_address	String	Connection address of Kafka Manager of the Kafka instance.
ssl_enable	Boolean	Whether security authentication is enabled. <ul style="list-style-type: none"> • true: enable • false: disabled
broker_ssl_enable	Boolean	Indicates whether to enable encrypted replica transmission among brokers. <ul style="list-style-type: none"> • true: enable • false: disable

Parameter	Type	Description
kafka_security_protocol	String	<p>Security protocol used by Kafka.</p> <p>If port-protocols is returned in the instance details, kafka_security_protocol works as the security protocol for private and public network access, and cross-VPC access.</p> <p>Otherwise, kafka_security_protocol works as the security protocol only for cross-VPC access. For the security protocol for private and public network access, see port_protocols.</p> <ul style="list-style-type: none"> • PLAINTEXT: The SSL certificate is not used for encrypted transmission, and username-password authentication is unavailable. This mode has higher performance but lower security. You are advised not to use this mode for public network access in the production environment. • SASL_SSL: Data is encrypted with SSL certificates for high-security transmission. • SASL_PLAINTEXT: Data is transmitted in plaintext with username and password authentication. This protocol uses the SCRAM-SHA-512 mechanism and delivers high performance.
sasl_enabled_mechanisms	Array of strings	<p>Authentication mechanism used after SASL is enabled.</p> <ul style="list-style-type: none"> • PLAIN: simple username and password verification. • SCRAM-SHA-512: user credential verification, which is more secure than PLAIN.
ssl_two_way_enable	Boolean	Indicates whether to enable two-way authentication.
cert_replaced	Boolean	Whether the certificate can be replaced.
public_management_connect_addresses	String	Address for accessing Kafka Manager over public networks.
enterprise_project_id	String	Enterprise project ID.

Parameter	Type	Description
is_logical_volume	Boolean	Whether the instance is a new instance. This parameter is used to distinguish old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> • true: New instance, which allows dynamic disk capacity expansion without restarting the instance. • false: Old instance.
extend_times	Integer	Number of disk expansion times. If the value exceeds 20, disk expansion is no longer allowed.
enable_auto_topic	Boolean	Whether automatic topic creation is enabled. <ul style="list-style-type: none"> • true: enabled • false: disabled
type	String	Instance type. The value can be cluster .
product_id	String	Product ID.
security_group_id	String	Security group ID.
security_group_name	String	Security group name.
subnet_id	String	Subnet ID.
available_zones	Array of strings	AZ to which the instance brokers belong. The AZ ID is returned.
available_zone_names	Array of strings	Name of the AZ to which the instance node belongs. The AZ name is returned.
total_storage_space	Integer	Message storage space in GB.
public_connect_address	String	Instance public access address. This parameter is available only when public access is enabled for the instance.
public_connect_domain_name	String	Public network access domain name of the instance. This parameter is available only when public access is enabled for the instance.
storage_resource_id	String	Storage resource ID.

Parameter	Type	Description
storage_spec_code	String	I/O specifications.
service_type	String	Service type.
storage_type	String	Storage class.
retention_policy	String	Message retention policy.
kafka_public_status	String	Whether public access is enabled for Kafka.
public_bandwidth	Integer	Public network access bandwidth.
enable_log_collection	Boolean	Whether log collection is enabled.
new_auth_cert	Boolean	Indicates whether to enable a new certificate.
cross_vpc_info	String	Cross-VPC access information.
ipv6_enable	Boolean	Whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	IPv6 connection address.
connector_enable	Boolean	Whether dumping is enabled. Dumping is not supported for the new specification type.
connector_node_number	Integer	Number of connectors.
connector_id	String	Dumping task ID.
rest_enable	Boolean	Whether Kafka REST is enabled.
rest_connect_address	String	Kafka REST connection address.
public_bandwidth	Integer	Public network access bandwidth. To be deleted.
message_query_instance_enable	Boolean	Whether message query is enabled.
vpc_client_plain	Boolean	Whether intra-VPC plaintext access is enabled.
support_features	String	List of features supported by the Kafka instance.
trace_enable	Boolean	Whether message tracing is enabled.
agent_enable	Boolean	Indicates whether the proxy is enabled.

Parameter	Type	Description
pod_connect_address	String	Connection address on the tenant side.
disk_encrypted	Boolean	Whether disk encryption is enabled.
disk_encrypted_key	String	Disk encryption key. If disk encryption is not enabled, this parameter is left blank.
kafka_private_connect_address	String	Private connection address of a Kafka instance.
kafka_private_connect_domain_name	String	Private connection domain name of a Kafka instance.
ces_version	String	Cloud Eye version.
public_access_enabled	String	Time when public access was enabled for an instance. Value range: <ul style="list-style-type: none"> • true: enabled • actived: disabled • closed: disabled • false: disabled
node_num	Integer	Node quantity.
port_protocols	PortProtocolsEntity object	Connection modes and addresses supported by an instance.
enable_acl	Boolean	Indicates whether access control is enabled.
new_spec_billing_enable	Boolean	Whether billing based on new specifications is enabled.
broker_num	Integer	Broker quantity.
tags	Array of TagEntity objects	Tag list.
dr_enable	Boolean	Indicates whether DR is enabled.

Table 5-11 PortProtocolsEntity

Parameter	Type	Description
private_plain_enable	Boolean	Whether the instance supports private plaintext access. <ul style="list-style-type: none"> • true: Yes • false: No
private_plain_address	String	Connection addresses of Kafka private plaintext access.
private_plain_domain_name	String	Private plaintext connection domain name.
private_sasl_ssl_enable	Boolean	Whether the instance supports private SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_ssl_address	String	Connection addresses of the Kafka private SASL_SSL access mode.
private_sasl_ssl_domain_name	String	Private SASL_SSL connection domain name.
private_sasl_plaintext_enable	Boolean	Whether the instance supports private SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_plaintext_address	String	Connection addresses of the Kafka private SASL_PLAINTEXT access mode.
private_sasl_plaintext_domain_name	String	Private SASL_PLAINTEXT connection domain name.
public_plain_enable	Boolean	Whether the instance supports public plaintext access. <ul style="list-style-type: none"> • true: Yes • false: No
public_plain_address	String	Connection addresses of Kafka public plaintext access.
public_plain_domain_name	String	Public plaintext connection domain name.
public_sasl_ssl_enable	Boolean	Whether the instance supports public SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No

Parameter	Type	Description
public_sasl_ssl_address	String	Connection addresses of the Kafka public SASL_SSL access mode.
public_sasl_ssl_domain_name	String	Public SASL_SSL connection domain name.
public_sasl_plaintext_enable	Boolean	Whether the instance supports public SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
public_sasl_plaintext_address	String	Connection addresses of the Kafka public SASL_PLAINTEXT access mode.
public_sasl_plaintext_domain_name	String	Public SASL_PLAINTEXT connection domain name.

Table 5-12 TagEntity

Parameter	Type	Description
key	String	Tag key. <ul style="list-style-type: none"> • Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code> • Cannot start with sys • Cannot start or end with a space.
value	String	Tag value. <ul style="list-style-type: none"> • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code>

Example Requests

Querying the instance list

```
GET https://{endpoint}/v2/{project_id}/instances
```

Example Responses

Status code: 200

Instance list queried.

```
{
  "instances": [ {
    "name": "kafka-2085975099",
    "engine": "kafka",
    "port": 9092,
    "status": "RUNNING",
    "type": "cluster",
    "specification": "100MB",
    "engine_version": "1.1.0",
    "connect_address": "192.168.0.100,192.168.0.61,192.168.0.72",
    "instance_id": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
    "charging_mode": 1,
    "vpc_id": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "vpc_name": "dms-test",
    "created_at": "1585618587087",
    "product_id": "00300-30308-0--0",
    "security_group_id": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "security_group_name": "Sys-default",
    "subnet_id": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "available_zones": [ "38b0f7a602344246bcb0da47b5d548e7" ],
    "available_zone_names": [ "AZ1" ],
    "user_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "user_name": "paas_dms",
    "maintain_begin": "02:00:00",
    "maintain_end": "06:00:00",
    "enable_log_collection": false,
    "new_auth_cert": false,
    "storage_space": 492,
    "total_storage_space": 600,
    "used_storage_space": 25,
    "partition_num": "300",
    "enable_publicip": false,
    "ssl_enable": false,
    "broker_ssl_enable": false,
    "cert_replaced": false,
    "management_connect_address": "https://192.168.0.100:9999",
    "cross_vpc_info": "{\n  \"192.168.0.61\":{\n    \"advertised_ip\": \"192.168.0.61\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx\", \"192.168.0.72\":{\n    \"advertised_ip\": \"192.168.0.72\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx\", \"192.168.0.100\":{\n    \"advertised_ip\": \"192.168.0.100\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx\"}}",
    "storage_resource_id": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "storage_spec_code": "dms.physical.storage.ultra",
    "service_type": "advanced",
    "storage_type": "hec",
    "enterprise_project_id": "0",
    "is_logical_volume": true,
    "extend_times": 0,
    "retention_policy": "produce_reject",
    "ipv6_enable": false,
    "ipv6_connect_addresses": [ ],
    "connector_enable": false,
    "connector_node_num": 0,
    "connector_id": "",
    "rest_enable": false,
    "rest_connect_address": "",
    "kafka_public_status": "closed",
    "public_bandwidth": 0,
    "message_query_inst_enable": true,
    "vpc_client_plain": false,
    "support_features":
      "kafka.new.pod.port.feature.physerver.kafka.topic.modify,feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.kafka.pulbic.dynamic,roma_app_enable,features.log.collection,auto_topic_switch,feature.physerver.kafka.user.manager",
    "trace_enable": false,
    "agent_enable": false,
    "pod_connect_address": "100.86.75.15:9080,100.86.142.77:9080,100.86.250.167:9080",
    "disk_encrypted": false,
    "kafka_private_connect_address": "192.168.0.61:9092,192.168.0.100:9092,192.168.0.72:9092",
  }
  ]
}
```



```
"enable_auto_topic" : false,  
"new_spec_billing_enable" : false,  
"ces_version" : "linux"  
}],  
"instance_num" : 1  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ListInstancesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListInstancesRequest request = new ListInstancesRequest();  
        try {  
            ListInstancesResponse response = client.listInstances(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstancesRequest()
        response = client.list_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstancesRequest{}
    response, err := client.ListInstances(request)
```

```

if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Instance list queried.

Error Codes

See [Error Codes](#).

5.1.3 Querying an Instance

Function

This API is used to query the details about a specified instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}

Table 5-13 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-14 Response body parameters

Parameter	Type	Description
name	String	Instance name.
engine	String	Message engine.
engine_version	String	Version.
description	String	Instance description
specification	String	Instance specifications.
storage_space	Integer	Message storage space in GB.
partition_num	String	Number of partitions in a Kafka instance.
used_storage_space	Integer	Used message storage space in GB.
dns_enable	Boolean	Indicates whether domain name access is enabled for an instance. <ul style="list-style-type: none"> • true: enable • false: disable
connect_address	String	IP address of an instance.
port	Integer	Port of an instance.
status	String	Instance status. For details, see Instance Status .
instance_id	String	Instance ID.
resource_spec_code	String	Resource specifications. <ul style="list-style-type: none"> • dms.instance.kafka.cluster.c3.mini: Kafka instance with 100 MB/s bandwidth • dms.instance.kafka.cluster.c3.small.2: Kafka instance with 300 MB/s bandwidth • dms.instance.kafka.cluster.c3.middle.2: Kafka instance with 600 MB/s bandwidth • dms.instance.kafka.cluster.c3.high.2: Kafka instance with 1200 MB/s bandwidth

Parameter	Type	Description
charging_mode	Integer	Billing mode. Options: 1 : pay-per-use; 0 : yearly/monthly.
vpc_id	String	VPC ID.
vpc_name	String	VPC name.
created_at	String	Time when the instance was created. The time is in the format of timestamp, that is, the offset milliseconds from 1970-01-01 00:00:00 UTC to the specified time.
subnet_name	String	Subnet name.
subnet_cidr	String	Subnet CIDR block.
user_id	String	User ID.
user_name	String	Username.
access_user	String	Username for accessing the instance.
order_id	String	Order ID. This parameter has a value only when the billing mode is yearly/monthly.
maintain_begin	String	Time at which the maintenance time window starts. The format is HH:mm:ss.
maintain_end	String	Time at which the maintenance time window ends. The format is HH:mm:ss.
enable_publicip	Boolean	Whether public access is enabled for the instance. <ul style="list-style-type: none"> • true: enabled • false: disabled
management_connect_address	String	Connection address of Kafka Manager of the Kafka instance.
ssl_enable	Boolean	Whether security authentication is enabled. <ul style="list-style-type: none"> • true: enable • false: disabled
broker_ssl_enable	Boolean	Indicates whether to enable encrypted replica transmission among brokers. <ul style="list-style-type: none"> • true: enable • false: disable

Parameter	Type	Description
kafka_security_protocol	String	<p>Security protocol used by Kafka.</p> <p>If port-protocols is returned in the instance details, kafka_security_protocol works as the security protocol for private and public network access, and cross-VPC access.</p> <p>Otherwise, kafka_security_protocol works as the security protocol only for cross-VPC access. For the security protocol for private and public network access, see port_protocols.</p> <ul style="list-style-type: none"> • PLAINTEXT: The SSL certificate is not used for encrypted transmission, and username-password authentication is unavailable. This mode has higher performance but lower security. You are advised not to use this mode for public network access in the production environment. • SASL_SSL: Data is encrypted with SSL certificates for high-security transmission. • SASL_PLAINTEXT: Data is transmitted in plaintext with username and password authentication. This protocol uses the SCRAM-SHA-512 mechanism and delivers high performance.
sasl_enabled_mechanisms	Array of strings	<p>Authentication mechanism used after SASL is enabled.</p> <ul style="list-style-type: none"> • PLAIN: simple username and password verification. • SCRAM-SHA-512: user credential verification, which is more secure than PLAIN.
ssl_two_way_enable	Boolean	Indicates whether to enable two-way authentication.
cert_replaced	Boolean	Whether the certificate can be replaced.
public_management_connect_addresses	String	Address for accessing Kafka Manager over public networks.
enterprise_project_id	String	Enterprise project ID.

Parameter	Type	Description
is_logical_volume	Boolean	Whether the instance is a new instance. This parameter is used to distinguish old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> • true: New instance, which allows dynamic disk capacity expansion without restarting the instance. • false: Old instance.
extend_times	Integer	Number of disk expansion times. If the value exceeds 20, disk expansion is no longer allowed.
enable_auto_topic	Boolean	Whether automatic topic creation is enabled. <ul style="list-style-type: none"> • true: enabled • false: disabled
type	String	Instance type. The value can be cluster .
product_id	String	Product ID.
security_group_id	String	Security group ID.
security_group_name	String	Security group name.
subnet_id	String	Subnet ID.
available_zones	Array of strings	AZ to which the instance brokers belong. The AZ ID is returned.
available_zone_names	Array of strings	Name of the AZ to which the instance node belongs. The AZ name is returned.
total_storage_space	Integer	Message storage space in GB.
public_connect_address	String	Instance public access address. This parameter is available only when public access is enabled for the instance.
public_connect_domain_name	String	Public network access domain name of the instance. This parameter is available only when public access is enabled for the instance.
storage_resource_id	String	Storage resource ID.

Parameter	Type	Description
storage_spec_code	String	I/O specifications.
service_type	String	Service type.
storage_type	String	Storage class.
retention_policy	String	Message retention policy.
kafka_public_status	String	Whether public access is enabled for Kafka.
public_bandwidth	Integer	Public network access bandwidth.
enable_log_collection	Boolean	Whether log collection is enabled.
new_auth_cert	Boolean	Indicates whether to enable a new certificate.
cross_vpc_info	String	Cross-VPC access information.
ipv6_enable	Boolean	Whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	IPv6 connection address.
connector_enable	Boolean	Whether dumping is enabled. Dumping is not supported for the new specification type.
connector_node_number	Integer	Number of connectors.
connector_id	String	Dumping task ID.
rest_enable	Boolean	Whether Kafka REST is enabled.
rest_connect_address	String	Kafka REST connection address.
public_bandwidth	Integer	Public network access bandwidth. To be deleted.
message_query_instance_enable	Boolean	Whether message query is enabled.
vpc_client_plain	Boolean	Whether intra-VPC plaintext access is enabled.
support_features	String	List of features supported by the Kafka instance.
trace_enable	Boolean	Whether message tracing is enabled.
agent_enable	Boolean	Indicates whether the proxy is enabled.

Parameter	Type	Description
pod_connect_address	String	Connection address on the tenant side.
disk_encrypted	Boolean	Whether disk encryption is enabled.
disk_encrypted_key	String	Disk encryption key. If disk encryption is not enabled, this parameter is left blank.
kafka_private_connect_address	String	Private connection address of a Kafka instance.
kafka_private_connect_domain_name	String	Private connection domain name of a Kafka instance.
ces_version	String	Cloud Eye version.
public_access_enabled	String	Time when public access was enabled for an instance. Value range: <ul style="list-style-type: none"> • true: enabled • actived: disabled • closed: disabled • false: disabled
node_num	Integer	Node quantity.
port_protocols	PortProtocolsEntity object	Connection modes and addresses supported by an instance.
enable_acl	Boolean	Indicates whether access control is enabled.
new_spec_billing_enable	Boolean	Whether billing based on new specifications is enabled.
broker_num	Integer	Broker quantity.
tags	Array of TagEntity objects	Tag list.
dr_enable	Boolean	Indicates whether DR is enabled.

Table 5-15 PortProtocolsEntity

Parameter	Type	Description
private_plain_enable	Boolean	Whether the instance supports private plaintext access. <ul style="list-style-type: none"> • true: Yes • false: No
private_plain_address	String	Connection addresses of Kafka private plaintext access.
private_plain_domain_name	String	Private plaintext connection domain name.
private_sasl_ssl_enable	Boolean	Whether the instance supports private SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_ssl_address	String	Connection addresses of the Kafka private SASL_SSL access mode.
private_sasl_ssl_domain_name	String	Private SASL_SSL connection domain name.
private_sasl_plaintext_enable	Boolean	Whether the instance supports private SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_plaintext_address	String	Connection addresses of the Kafka private SASL_PLAINTEXT access mode.
private_sasl_plaintext_domain_name	String	Private SASL_PLAINTEXT connection domain name.
public_plain_enable	Boolean	Whether the instance supports public plaintext access. <ul style="list-style-type: none"> • true: Yes • false: No
public_plain_address	String	Connection addresses of Kafka public plaintext access.
public_plain_domain_name	String	Public plaintext connection domain name.
public_sasl_ssl_enable	Boolean	Whether the instance supports public SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No

Parameter	Type	Description
public_sasl_ssl_address	String	Connection addresses of the Kafka public SASL_SSL access mode.
public_sasl_ssl_domain_name	String	Public SASL_SSL connection domain name.
public_sasl_plaintext_enable	Boolean	Whether the instance supports public SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
public_sasl_plaintext_address	String	Connection addresses of the Kafka public SASL_PLAINTEXT access mode.
public_sasl_plaintext_domain_name	String	Public SASL_PLAINTEXT connection domain name.

Table 5-16 TagEntity

Parameter	Type	Description
key	String	Tag key. <ul style="list-style-type: none"> • Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code> • Cannot start with sys • Cannot start or end with a space.
value	String	Tag value. <ul style="list-style-type: none"> • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code>

Example Requests

Querying an instance

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

Example Responses

Status code: 200

Specified instance queried.

```
{
  "name": "kafka-2085975099",
  "engine": "kafka",
  "port": 9092,
  "status": "RUNNING",
  "type": "cluster",
  "specification": "100MB",
  "engine_version": "1.1.0",
  "connect_address": "192.168.0.100,192.168.0.61,192.168.0.72",
  "instance_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
  "charging_mode": 1,
  "vpc_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "vpc_name": "dms-test",
  "created_at": "1585618587087",
  "product_id": "00300-30308-0--0",
  "security_group_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "security_group_name": "Sys-default",
  "subnet_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "available_zones": [ "38b0f7a602344246bcb0da47b5d548e7" ],
  "available_zone_names": [ "AZ1" ],
  "user_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "user_name": "paas_dms",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00",
  "enable_log_collection": false,
  "new_auth_cert": false,
  "storage_space": 492,
  "total_storage_space": 600,
  "used_storage_space": 25,
  "partition_num": "300",
  "enable_publicip": false,
  "ssl_enable": false,
  "broker_ssl_enable": false,
  "cert_replaced": false,
  "management_connect_address": "https://192.168.0.100:9999",
  "cross_vpc_info": "{\"192.168.0.61\":{\"advertised_ip\":\"192.168.0.61\",\"port\":\"9011\",\"port_id\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"},\"192.168.0.72\":{\"advertised_ip\":\"192.168.0.72\",\"port\":\"9011\",\"port_id\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"},\"192.168.0.100\":{\"advertised_ip\":\"192.168.0.100\",\"port\":\"9011\",\"port_id\":\"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"}}",
  "storage_resource_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "storage_spec_code": "dms.physical.storage.ultra",
  "service_type": "advanced",
  "storage_type": "hec",
  "enterprise_project_id": "0",
  "is_logical_volume": true,
  "extend_times": 0,
  "retention_policy": "produce_reject",
  "ipv6_enable": false,
  "ipv6_connect_addresses": [ ],
  "connector_enable": false,
  "connector_node_num": 0,
  "connector_id": "",
  "rest_enable": false,
  "rest_connect_address": "",
  "kafka_public_status": "closed",
  "public_bandwidth": 0,
  "message_query_inst_enable": true,
  "vpc_client_plain": false,
  "support_features": "kafka.new.pod.port.feature.physerver.kafka.topic.modify,feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.kafka.publlic.dynamic,roma_app_enable,features.log.collection,auto_topic_switch,feature.physerver.kafka.user.manager",
  "trace_enable": false,
  "agent_enable": false,
  "pod_connect_address": "100.86.75.15:9080,100.86.142.77:9080,100.86.250.167:9080",
  "disk_encrypted": false,
  "kafka_private_connect_address": "192.168.0.61:9092,192.168.0.100:9092,192.168.0.72:9092",
  "enable_auto_topic": false,

```

```
"new_spec_billing_enable" : false,
"ces_version" : "linux",
"port_protocols" : "{ \"private_plain_enable\" : true, \"private_plain_address\" :
\\\"192.xxx.xxx.xxx:9092,192.xxx.xxx.xxx:9092,192.xxx.xxx.xxx:9092\\\", \"private_sasl_ssl_enable\" :
true, \"private_sasl_ssl_address\" :
\\\"192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093\\\", \"private_sasl_plaintext_enable\" :
false, \"private_sasl_plaintext_address\" : \\\"\\\", \"public_plain_enable\" : true, \"public_plain_address\" :
\\\"100.xxx.xxx.xxx:9094,100.xxx.xxx.xxx:9094,100.xxx.xxx.xxx:9094\\\", \"public_sasl_ssl_enable\" :
true, \"public_sasl_ssl_address\" :
\\\"100.xxx.xxx.xxx:9095,100.xxx.xxx.xxx:9095,100.xxx.xxx.xxx:9095\\\", \"public_sasl_plaintext_enable\" :
false, \"public_sasl_plaintext_address\" : \\\"\\\" }"
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceRequest request = new ShowInstanceRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowInstanceResponse response = client.showInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkkafka.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = KafkaClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ShowInstanceRequest()  
        request.instance_id = "{instance_id}"  
        response = client.show_instance(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```

```

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Specified instance queried.

Error Codes

See [Error Codes](#).

5.1.4 Deleting an Instance

Function

This API is used to delete an instance to release all the resources occupied by it.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}

Table 5-17 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.

Request Parameters

None

Response Parameters

None

Example Requests

Deleting an instance

```
DELETE https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
```



```
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
DeleteInstanceRequest request = new DeleteInstanceRequest();
request.withInstanceId("{instance_id}");
try {
    DeleteInstanceResponse response = client.deleteInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteInstanceRequest()
        request.instance_id = "{instance_id}"
        response = client.delete_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
```

```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteInstanceRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.DeleteInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The specified instance is deleted successfully.

Error Codes

See [Error Codes](#).

5.1.5 Modifying Instance Information

Function

This API is used to modify the instance information.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}

Table 5-18 Path Parameters

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

Request Parameters

Table 5-19 Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Instance name. An instance name starts with a letter, consists of 4 to 64 characters, and can contain only letters, digits, underscores (_), and hyphens (-).
description	No	String	Description of an instance. The description supports up to 1024 characters. NOTE The backslash (\) and quotation mark (") are special characters for JSON messages. When using these characters in a parameter value, add the escape character (\) before the characters, for example, \ and \".
maintain_begin	No	String	Time at which the maintenance window starts. The format is HH:mm:ss.
maintain_end	No	String	Time at which the maintenance window ends. The format is HH:mm:ss.
security_group_id	No	String	Security group ID. To obtain it, log in to the VPC console and view the security group ID on the security group details page.

Parameter	Mandatory	Type	Description
retention_policy	No	String	Capacity threshold policy. Options: <ul style="list-style-type: none"> • produce_reject: New messages cannot be created. • time_base: The earliest messages are deleted.
enterprise_project_id	No	String	Enterprise project.

Response Parameters

None

Example Requests

- Modifying the name and description of an instance.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

```
{
  "name": "kafka001",
  "description": "kafka description"
}
```

- Modifying the name, description, and maintenance time window of an instance.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

```
{
  "name": "dms002",
  "description": "instance description",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00"
}
```

- Changing the capacity threshold policy.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}
```

```
{
  "retention_policy": "time_base"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

- Modifying the name and description of an instance.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withDescription("kafka description");
        body.withName("kafka001");
        request.withBody(body);
        try {
            UpdateInstanceResponse response = client.updateInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Modifying the name, description, and maintenance time window of an instance.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withMaintainEnd("06:00:00");
        body.withMaintainBegin("02:00:00");
        body.withDescription("instance description");
        body.withName("dms002");
        request.withBody(body);
        try {
            UpdateInstanceResponse response = client.updateInstance(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Changing the capacity threshold policy.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateInstanceRequest request = new UpdateInstanceRequest();
request.withInstanceId("{instance_id}");
UpdateInstanceReq body = new UpdateInstanceReq();
body.withRetentionPolicy(UpdateInstanceReq.RetentionPolicyEnum.fromValue("time_base"));
request.withBody(body);
try {
    UpdateInstanceResponse response = client.updateInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Modifying the name and description of an instance.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```
request = UpdateInstanceRequest()
request.instance_id = "{instance_id}"
request.body = UpdateInstanceReq(
    description="kafka description",
    name="kafka001"
)
response = client.update_instance(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Modifying the name, description, and maintenance time window of an instance.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateInstanceReq(
            maintain_end="06:00:00",
            maintain_begin="02:00:00",
            description="instance description",
            name="dms002"
        )
        response = client.update_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Changing the capacity threshold policy.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
```



```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateInstanceRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdateInstanceReq(
        retention_policy="time_base"
    )
    response = client.update_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Modifying the name and description of an instance.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{
        request.InstanceId = "{instance_id}"
```

```
descriptionUpdateInstanceReq:= "kafka description"
nameUpdateInstanceReq:= "kafka001"
request.Body = &model.UpdateInstanceReq{
    Description: &descriptionUpdateInstanceReq,
    Name: &nameUpdateInstanceReq,
}
response, err := client.UpdateInstance(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Modifying the name, description, and maintenance time window of an instance.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{
        request.InstanceId = "{instance_id}"
        maintainEndUpdateInstanceReq:= "06:00:00"
        maintainBeginUpdateInstanceReq:= "02:00:00"
        descriptionUpdateInstanceReq:= "instance description"
        nameUpdateInstanceReq:= "dms002"
        request.Body = &model.UpdateInstanceReq{
            MaintainEnd: &maintainEndUpdateInstanceReq,
            MaintainBegin: &maintainBeginUpdateInstanceReq,
            Description: &descriptionUpdateInstanceReq,
            Name: &nameUpdateInstanceReq,
        }
    }
    response, err := client.UpdateInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Changing the capacity threshold policy.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{}
    request.InstanceId = "{instance_id}"
    retentionPolicyUpdateInstanceReq:=
model.GetUpdateInstanceReqRetentionPolicyEnum().TIME_BASE
    request.Body = &model.UpdateInstanceReq{
        RetentionPolicy: &retentionPolicyUpdateInstanceReq,
    }
    response, err := client.UpdateInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The instance information is modified successfully.

Error Codes

See [Error Codes](#).

5.1.6 Batch Restarting or Deleting Instances

Function

This API is used to restart or delete instances in batches.

When an instance is being restarted, message retrieval and creation requests of the client will be rejected.

Deleting an instance will delete the data in the instance without any backup. Exercise caution when performing this operation.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/action

Table 5-20 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .

Request Parameters

Table 5-21 Request body parameters

Parameter	Mandatory	Type	Description
instances	No	Array of strings	List of instance IDs.
action	Yes	String	Operation to be performed on instances. The value can be restart or delete .
all_failure	No	String	Value kafka indicates all Kafka instances that fail to be created are to be deleted.

Response Parameters

Status code: 200

Table 5-22 Response body parameters

Parameter	Type	Description
results	Array of results objects	Result of instance modification.

Table 5-23 results

Parameter	Type	Description
result	String	Operation result. <ul style="list-style-type: none"> • success: The operation succeeded. • failed: The operation failed.
instance	String	Instance ID.

Example Requests

- Restarting instances in batches.

```
POST https://{endpoint}/v2/{project_id}/instances/action
```

```
{
  "action": "restart",
  "instances": [ "54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640" ]
}
```

- Deleting instances in batches.

```
POST https://{endpoint}/v2/{project_id}/instances/action
```

```
{
  "action": "delete",
  "instances": [ "54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-
dbad-4d79-9610-7163e6f8b640" ]
}
```

- Deleting all instances that fail to be created.

```
POST https://{endpoint}/v2/{project_id}/instances/action
```

```
{
  "action": "delete",
  "all_failure": "kafka"
}
```

Example Responses

Status code: 200

The instances are restarted or deleted successfully.

```
{
  "results": [ {
```

```
"result" : "success",  
"instance" : "019cacb7-4ff0-4d3c-9f33-f5f7b7fdc0e6"  
} ]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Restarting instances in batches.

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class BatchRestartOrDeleteInstancesSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
  
        BatchRestartOrDeleteInstancesRequest request = new BatchRestartOrDeleteInstancesRequest();  
        BatchRestartOrDeleteInstanceReq body = new BatchRestartOrDeleteInstanceReq();  
        List<String> listbodyInstances = new ArrayList<>();  
        listbodyInstances.add("54602a9d-5e22-4239-9123-77e350df4a34");  
        listbodyInstances.add("7166cdea-dbad-4d79-9610-7163e6f8b640");  
        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("restart"));  
        body.withInstances(listbodyInstances);  
        request.withBody(body);  
        try {  
            BatchRestartOrDeleteInstancesResponse response =  
client.batchRestartOrDeleteInstances(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
        }  
    }  
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- Deleting instances in batches.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchRestartOrDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchRestartOrDeleteInstancesRequest request = new BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new BatchRestartOrDeleteInstanceReq();
        List<String> listbodyInstances = new ArrayList<>();
        listbodyInstances.add("54602a9d-5e22-4239-9123-77e350df4a34");
        listbodyInstances.add("7166cdea-dbad-4d79-9610-7163e6f8b640");
        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("delete"));
        body.withInstances(listbodyInstances);
        request.withBody(body);
        try {
            BatchRestartOrDeleteInstancesResponse response =
client.batchRestartOrDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- Deleting all instances that fail to be created.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class BatchRestartOrDeleteInstancesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchRestartOrDeleteInstancesRequest request = new BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new BatchRestartOrDeleteInstanceReq();
        body.withAllFailure(BatchRestartOrDeleteInstanceReq.AllFailureEnum.fromValue("kafka"));
        body.withAction(BatchRestartOrDeleteInstanceReq.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchRestartOrDeleteInstancesResponse response =
            client.batchRestartOrDeleteInstances(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- Restarting instances in batches.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *
```



```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchRestartOrDeleteInstancesRequest()
        listInstancesbody = [
            "54602a9d-5e22-4239-9123-77e350df4a34",
            "7166cdea-dbad-4d79-9610-7163e6f8b640"
        ]
        request.body = BatchRestartOrDeleteInstanceReq(
            action="restart",
            instances=listInstancesbody
        )
        response = client.batch_restart_or_delete_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Deleting instances in batches.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchRestartOrDeleteInstancesRequest()
        listInstancesbody = [
            "54602a9d-5e22-4239-9123-77e350df4a34",
            "7166cdea-dbad-4d79-9610-7163e6f8b640"
        ]
    ]
```

```
request.body = BatchRestartOrDeleteInstanceReq(
    action="delete",
    instances=listInstancesbody
)
response = client.batch_restart_or_delete_instances(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- Deleting all instances that fail to be created.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchRestartOrDeleteInstancesRequest()
        request.body = BatchRestartOrDeleteInstanceReq(
            all_failure="kafka",
            action="delete"
        )
        response = client.batch_restart_or_delete_instances(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Restarting instances in batches.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchRestartOrDeleteInstancesRequest{}
var listInstancesbody = []string{
    "54602a9d-5e22-4239-9123-77e350df4a34",
    "7166cdea-dbad-4d79-9610-7163e6f8b640",
}
request.Body = &model.BatchRestartOrDeleteInstanceReq{
    Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().RESTART,
    Instances: &listInstancesbody,
}
response, err := client.BatchRestartOrDeleteInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Deleting instances in batches.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
}
```

```
request := &model.BatchRestartOrDeleteInstancesRequest{}
var listInstancesbody = []string{
    "54602a9d-5e22-4239-9123-77e350df4a34",
    "7166cdea-dbad-4d79-9610-7163e6f8b640",
}
request.Body = &model.BatchRestartOrDeleteInstanceReq{
    Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().DELETE,
    Instances: &listInstancesbody,
}
response, err := client.BatchRestartOrDeleteInstances(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Deleting all instances that fail to be created.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRestartOrDeleteInstancesRequest{}
    allFailureBatchRestartOrDeleteInstanceReq:=
    model.GetBatchRestartOrDeleteInstanceReqAllFailureEnum().KAFKA
    request.Body = &model.BatchRestartOrDeleteInstanceReq{
        AllFailure: &allFailureBatchRestartOrDeleteInstanceReq,
        Action: model.GetBatchRestartOrDeleteInstanceReqActionEnum().DELETE,
    }
    response, err := client.BatchRestartOrDeleteInstances(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The instances are restarted or deleted successfully.
204	All Kafka instances that fail to be created are deleted successfully.

Error Codes

See [Error Codes](#).

5.1.7 Obtaining Instance Configurations

Function

This API is used to obtain instance configurations.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/configs

Table 5-24 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-25 Response body parameters

Parameter	Type	Description
kafka_configs	Array of InstanceConfig objects	Kafka configurations.

Table 5-26 InstanceConfig

Parameter	Type	Description
name	String	Configuration name.
valid_values	String	Valid value.
default_value	String	Default value.
config_type	String	Configuration type. The value can be static or dynamic .
value	String	Current value.
value_type	String	Value type.

Example Requests

GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/configs

Example Responses

Status code: 200

Configuration obtained.

```
{
  "kafka_configs": [ {
    "name": "min.insync.replicas",
    "valid_values": "1~3",
    "default_value": "1",
    "config_type": "dynamic",
    "value": "1",
    "value_type": "integer"
  }, {
    "name": "message.max.bytes",
    "valid_values": "0~10485760",
    "default_value": "10485760",
    "config_type": "dynamic",
    "value": "10485760",
    "value_type": "integer"
  }, {
    "name": "auto.create.groups.enable",
    "valid_values": "true,false",
    "default_value": "true",
    "config_type": "dynamic",
    "value": "true",
    "value_type": "enum"
  }, {
    "name": "connections.max.idle.ms",
```

```
"valid_values" : "5000~600000",
"default_value" : "600000",
"config_type" : "static",
"value" : "600000",
"value_type" : "integer"
}, {
  "name" : "log.retention.hours",
  "valid_values" : "1~168",
  "default_value" : "72",
  "config_type" : "static",
  "value" : "72",
  "value_type" : "integer"
}, {
  "name" : "max.connections.per.ip",
  "valid_values" : "100~20000",
  "default_value" : "1000",
  "config_type" : "dynamic",
  "value" : "1000",
  "value_type" : "integer"
}, {
  "name" : "group.max.session.timeout.ms",
  "valid_values" : "6000~1800000",
  "default_value" : "1800000",
  "config_type" : "static",
  "value" : "1800000",
  "value_type" : "integer"
}, {
  "name" : "unclean.leader.election.enable",
  "valid_values" : "true,false",
  "default_value" : "false",
  "config_type" : "dynamic",
  "value" : "false",
  "value_type" : "enum"
}, {
  "name" : "default.replication.factor",
  "valid_values" : "1~3",
  "default_value" : "3",
  "config_type" : "static",
  "value" : "3",
  "value_type" : "integer"
}, {
  "name" : "offsets.retention.minutes",
  "valid_values" : "1440~30240",
  "default_value" : "20160",
  "config_type" : "dynamic",
  "value" : "20160",
  "value_type" : "integer"
}, {
  "name" : "num.partitions",
  "valid_values" : "1~200",
  "default_value" : "3",
  "config_type" : "static",
  "value" : "3",
  "value_type" : "integer"
}, {
  "name" : "group.min.session.timeout.ms",
  "valid_values" : "6000~300000",
  "default_value" : "6000",
  "config_type" : "static",
  "value" : "6000",
  "value_type" : "integer"
}, {
  "name" : "allow.everyone.if.no.acl.found",
  "valid_values" : "true,false",
  "default_value" : "true",
  "config_type" : "static",
  "value" : "true",
  "value_type" : "enum"
}
```

```
    }]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ShowInstanceConfigsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowInstanceConfigsRequest request = new ShowInstanceConfigsRequest();  
        request.withInstanceId("{instance_id}");  
        try {  
            ShowInstanceConfigsResponse response = client.showInstanceConfigs(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```



```
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceConfigsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_configs(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceConfigsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceConfigs(request)
```

```

if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Configuration obtained.

Error Codes

See [Error Codes](#).

5.1.8 Modifying Instance Configurations

Function

This API is used to modify instance configurations.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/configs

Table 5-27 Path Parameters

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

Request Parameters

Table 5-28 Request body parameters

Parameter	Mandatory	Type	Description
kafka_configs	No	Array of ModifyInstanceConfig objects	Configurations to be modified.

Table 5-29 ModifyInstanceConfig

Parameter	Mandatory	Type	Description
name	No	String	Names of configurations to be modified.
value	No	String	New value of the modified configuration.

Response Parameters

Status code: 200

Table 5-30 Response body parameters

Parameter	Type	Description
job_id	String	Configuration modification task ID.
dynamic_config	Integer	Number of dynamic configuration parameters to be modified.
static_config	Integer	Number of static configuration parameters to be modified.

Example Requests

Modifying the threshold for idle connection timeout and the log deletion interval.

PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/configs

```
{
  "kafka_configs": [ {
    "name": "connections.max.idle.ms",
    "value": "500000"
  }, {
    "name": "log.retention.hours",
    "value": "66"
  } ]
}
```

Example Responses

Status code: 200

Configuration modified.

```
{
  "job_id" : "8abfa7b38ba79a20018ba9afc550576a",
  "dynamic_config" : 0,
  "static_config" : 2
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying the threshold for idle connection timeout and the log deletion interval.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ModifyInstanceConfigsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ModifyInstanceConfigsRequest request = new ModifyInstanceConfigsRequest();
        request.withInstanceId("{instance_id}");
        ModifyInstanceConfigsReq body = new ModifyInstanceConfigsReq();
        List<ModifyInstanceConfig> listbodyKafkaConfigs = new ArrayList<>();
        listbodyKafkaConfigs.add(
            new ModifyInstanceConfig()
                .withName("connections.max.idle.ms")
                .withValue("500000")
        );
        listbodyKafkaConfigs.add(
            new ModifyInstanceConfig()
                .withName("log.retention.hours")
```

```
        .withValue("66")
    );
    body.withKafkaConfigs(listbodyKafkaConfigs);
    request.withBody(body);
    try {
        ModifyInstanceConfigsResponse response = client.modifyInstanceConfigs(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Modifying the threshold for idle connection timeout and the log deletion interval.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ModifyInstanceConfigsRequest()
        request.instance_id = "{instance_id}"
        listKafkaConfigsbody = [
            ModifyInstanceConfig(
                name="connections.max.idle.ms",
                value="500000"
            ),
            ModifyInstanceConfig(
                name="log.retention.hours",
                value="66"
            )
        ]
        request.body = ModifyInstanceConfigsReq(
            kafka_configs=listKafkaConfigsbody
        )
        response = client.modify_instance_configs(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

Modifying the threshold for idle connection timeout and the log deletion interval.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ModifyInstanceConfigsRequest{}
    request.InstanceId = "{instance_id}"
    nameKafkaConfigs:= "connections.max.idle.ms"
    valueKafkaConfigs:= "500000"
    nameKafkaConfigs1:= "log.retention.hours"
    valueKafkaConfigs1:= "66"
    var listKafkaConfigbody = []model.ModifyInstanceConfig{
        {
            Name: &nameKafkaConfigs,
            Value: &valueKafkaConfigs,
        },
        {
            Name: &nameKafkaConfigs1,
            Value: &valueKafkaConfigs1,
        },
    }
    request.Body = &model.ModifyInstanceConfigsReq{
        KafkaConfigs: &listKafkaConfigbody,
    }
    response, err := client.ModifyInstanceConfigs(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Configuration modified.

Error Codes

See [Error Codes](#).

5.1.9 Upgrading an Instance

Function

This API is used to upgrading the instance kernel.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/upgrade

Table 5-31 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-32 Request body parameters

Parameter	Mandatory	Type	Description
is_schedule	No	Boolean	Whether the task is scheduled. If no, is_schedule and execute_at can be left blank. If yes, is_schedule is true and execute_at must be specified.
execute_at	No	Long	Schedule time. The value is a UNIX timestamp, in ms.

Response Parameters

Status code: 200

Table 5-33 Response body parameters

Parameter	Type	Description
-	String	ID of the submitted upgrade task.

Status code: 400

Table 5-34 Response body parameters

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

Example Requests

Upgrading an instance whose ID is *instance_id*

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/upgrade
{
  "is_schedule": true,
  "execute_at": 1695796358970
}
```

Example Responses

Status code: 200

Successful

93b94287-728d-4bb1-a158-cb66cb0854e7

Status Codes

Status Code	Description
200	Successful
400	Failed

Error Codes

See [Error Codes](#).

5.2 Instance Management

5.2.1 Resetting the Password

Function

This API is used to reset the password for an instance with SSL enabled.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/password

Table 5-35 Path Parameters

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

Request Parameters

Table 5-36 Request body parameters

Parameter	Mandatory	Type	Description
new_password	Yes	String	<p>The password can contain 8 to 32 characters, and must contain at least three types of the following characters:</p> <ul style="list-style-type: none"> • Uppercase letters • Lowercase letters • Digits • Special characters `~!@#\$%^&*()-_+=\ [{]}:;'"",<.>/?` and spaces, and cannot start with a hyphen (-).

Response Parameters

None

Example Requests

Resetting a password

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/password
{
  "new_password" : "*****"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Resetting a password

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class ResetPasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetPasswordRequest request = new ResetPasswordRequest();
        request.withInstanceId("{instance_id}");
        ResetPasswordReq body = new ResetPasswordReq();
        body.withNewPassword("*****");
        request.withBody(body);
        try {
            ResetPasswordResponse response = client.resetPassword(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Resetting a password

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = KafkaClient.new_builder() \  
  .with_credentials(credentials) \  
  .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
  .build()  
  
try:  
  request = ResetPasswordRequest()  
  request.instance_id = "{instance_id}"  
  request.body = ResetPasswordReq(  
    new_password="*****"  
  )  
  response = client.reset_password(request)  
  print(response)  
except exceptions.ClientRequestException as e:  
  print(e.status_code)  
  print(e.request_id)  
  print(e.error_code)  
  print(e.error_msg)
```

Go

Resetting a password

```
package main  
  
import (  
  "fmt"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
  kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
  region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
  // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
  // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
  // variables and decrypted during use to ensure security.  
  // In this example, AK and SK are stored in environment variables for authentication. Before running this  
  // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
  ak := os.Getenv("CLOUD_SDK_AK")  
  sk := os.Getenv("CLOUD_SDK_SK")  
  projectId := "{project_id}"  
  
  auth := basic.NewCredentialsBuilder().  
    WithAk(ak).  
    WithSk(sk).  
    WithProjectId(projectId).  
    Build()  
  
  client := kafka.NewKafkaClient(  
    kafka.KafkaClientBuilder().  
      WithRegion(region.ValueOf("<YOUR REGION>")).  
      WithCredential(auth).  
      Build())  
  
  request := &model.ResetPasswordRequest{}  
  request.InstanceId = "{instance_id}"  
  request.Body = &model.ResetPasswordReq{  
    NewPassword: "*****",  
  }  
  response, err := client.ResetPassword(request)  
  if err == nil {  
    fmt.Printf("%+v\n", response)  
  } else {  
    fmt.Println(err)  
  }  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The password is reset successfully.

Error Codes

See [Error Codes](#).

5.2.2 Resetting Kafka Manager Password

Function

This API is used to reset the Kafka Manager password.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/kafka-manager-password

Table 5-37 Path Parameters

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

Request Parameters

Table 5-38 Request body parameters

Parameter	Mandatory	Type	Description
new_password	No	String	The password can contain 8 to 32 characters, and must contain at least three types of the following characters: <ul style="list-style-type: none"> • Uppercase letters • Lowercase letters • Digits • Special characters `~!@#\$%^&*()-_+=\ []{};:'"''<,.>/?` and spaces, and cannot start with a hyphen (-).

Response Parameters

None

Example Requests

Resetting Kafka Manager password

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/kafka-manager-password
{
  "new_password" : "*****"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Resetting Kafka Manager password

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class ResetManagerPasswordSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetManagerPasswordRequest request = new ResetManagerPasswordRequest();
        request.withInstanceId("{instance_id}");
        ResetManagerPasswordReq body = new ResetManagerPasswordReq();
        body.withNewPassword("*****");
        request.withBody(body);
        try {
            ResetManagerPasswordResponse response = client.resetManagerPassword(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Resetting Kafka Manager password

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = KafkaClient.new_builder() \  
  .with_credentials(credentials) \  
  .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
  .build()  
  
try:  
  request = ResetManagerPasswordRequest()  
  request.instance_id = "{instance_id}"  
  request.body = ResetManagerPasswordReq(  
    new_password="*****"  
  )  
  response = client.reset_manager_password(request)  
  print(response)  
except exceptions.ClientRequestException as e:  
  print(e.status_code)  
  print(e.request_id)  
  print(e.error_code)  
  print(e.error_msg)
```

Go

Resetting Kafka Manager password

```
package main  
  
import (  
  "fmt"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
  kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
  region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
  // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
  // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
  // variables and decrypted during use to ensure security.  
  // In this example, AK and SK are stored in environment variables for authentication. Before running this  
  // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
  ak := os.Getenv("CLOUD_SDK_AK")  
  sk := os.Getenv("CLOUD_SDK_SK")  
  projectId := "{project_id}"  
  
  auth := basic.NewCredentialsBuilder().  
    WithAk(ak).  
    WithSk(sk).  
    WithProjectId(projectId).  
    Build()  
  
  client := kafka.NewKafkaClient(  
    kafka.KafkaClientBuilder().  
      WithRegion(region.ValueOf("<YOUR REGION>")).  
      WithCredential(auth).  
      Build())  
  
  request := &model.ResetManagerPasswordRequest{}  
  request.InstanceId = "{instance_id}"  
  newPasswordResetManagerPasswordReq:= "*****"  
  request.Body = &model.ResetManagerPasswordReq{  
    NewPassword: &newPasswordResetManagerPasswordReq,  
  }  
  response, err := client.ResetManagerPassword(request)  
  if err == nil {  
    fmt.Printf("%+v\n", response)  
  } else {  
    fmt.Println(err)  
  }  
}
```


More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The password is reset successfully.

Error Codes

See [Error Codes](#).

5.2.3 Restarting Kafka Manager

Function

This API is used to restart Kafka Manager.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/restart-kafka-manager

Table 5-39 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-40 Response body parameters

Parameter	Type	Description
result	String	Execution result.
instance_id	String	Instance ID.

Example Requests

Restarting Kafka Manager

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/restart-kafka-manager
```

Example Responses

Status code: 200

Kafka Manager is restarted successfully.

```
{
  "result" : "success",
  "instance_id" : "605cd78c-92dc-4335-8bae-43677f31fd6c"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class RestartManagerSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
RestartManagerRequest request = new RestartManagerRequest();
request.withInstanceId("{instance_id}");
try {
    RestartManagerResponse response = client.restartManager(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = RestartManagerRequest()
        request.instance_id = "{instance_id}"
        response = client.restart_manager(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
```

```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartManagerRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.RestartManager(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Kafka Manager is restarted successfully.

Error Codes

See [Error Codes](#).

5.2.4 Configuring Automatic Topic Creation

Function

This API is used to enable or disable automatic topic creation.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/autotopic

Table 5-41 Path Parameters

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

Request Parameters

Table 5-42 Request body parameters

Parameter	Mandatory	Type	Description
enable_auto_t opic	Yes	Boolean	Whether to enable automatic topic creation.

Response Parameters

None

Example Requests

Enabling automatic topic creation.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/autotopic
{
  "enable_auto_topic" : true
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Enabling automatic topic creation.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceAutoCreateTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceAutoCreateTopicRequest request = new UpdateInstanceAutoCreateTopicRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceAutoCreateTopicReq body = new UpdateInstanceAutoCreateTopicReq();
        body.withEnableAutoTopic(true);
        request.withBody(body);
        try {
            UpdateInstanceAutoCreateTopicResponse response = client.updateInstanceAutoCreateTopic(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Enabling automatic topic creation.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateInstanceAutoCreateTopicRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdateInstanceAutoCreateTopicReq(
        enable_auto_topic=True
    )
    response = client.update_instance_auto_create_topic(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Enabling automatic topic creation.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceAutoCreateTopicRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.UpdateInstanceAutoCreateTopicReq{
        EnableAutoTopic: true,
    }
    response, err := client.UpdateInstanceAutoCreateTopic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    }
}
```

```

} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The function is enabled or disabled successfully.

Error Codes

See [Error Codes](#).

5.2.5 Modifying the Private IP Address for Cross-VPC Access

Function

This API is used to modify the private IP address for cross-VPC access.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/crossvpc/modify

Table 5-43 Path Parameters

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

Request Parameters

Table 5-44 Request body parameters

Parameter	Mandatory	Type	Description
advertised_ip_contents	Yes	Map<String,String>	<p>User-defined advertised_ip_contents key-value pair.</p> <p>The key is the listeners IP address.</p> <p>The value is the advertised.listeners IP address or domain name.</p> <p>NOTE Fill in the items that are not modified during IP address change.</p>

Response Parameters

Status code: 200

Table 5-45 Response body parameters

Parameter	Type	Description
success	Boolean	Result of the cross-VPC access modification.
results	Array of results objects	Details of the result of the cross-VPC access modification.

Table 5-46 results

Parameter	Type	Description
advertised_ip	String	advertised.listeners IP address or domain name.
success	Boolean	Status of the cross-VPC access modification.
ip	String	Listeners IP address.

Example Requests

Modifying the private IP address for cross-VPC access.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/crossvpc/modify

{
  "advertised_ip_contents": {
    "192.168.245.246": "192.168.245.247",
    "192.168.197.36": "192.168.197.38",
    "192.168.190.11": "192.168.190.11"
  }
}
```

Example Responses

Status code: 200

The private IP address for cross-VPC access is modified successfully.

```
{
  "success": true,
  "results": [ {
    "advertised_ip": "192.168.197.36",
    "success": true,
    "ip": "192.168.197.36"
  }, {
    "advertised_ip": "192.168.190.11",
    "success": true,
    "ip": "192.168.190.11"
  }, {
    "advertised_ip": "192.168.245.255",
    "success": true,
    "ip": "192.168.245.246"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying the private IP address for cross-VPC access.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateInstanceCrossVpcIpSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateInstanceCrossVpcIpRequest request = new UpdateInstanceCrossVpcIpRequest();
request.withInstanceId("{instance_id}");
UpdateInstanceCrossVpcIpReq body = new UpdateInstanceCrossVpcIpReq();
Map<String, String> listbodyAdvertisedIpContents = new HashMap<>();
listbodyAdvertisedIpContents.put("192.168.245.246", "192.168.245.247");
listbodyAdvertisedIpContents.put("192.168.197.36", "192.168.197.38");
listbodyAdvertisedIpContents.put("192.168.190.11", "192.168.190.11");
body.withAdvertisedIpContents(listbodyAdvertisedIpContents);
request.withBody(body);
try {
    UpdateInstanceCrossVpcIpResponse response = client.updateInstanceCrossVpcIp(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

Python

Modifying the private IP address for cross-VPC access.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceCrossVpcIpRequest()
        request.instance_id = "{instance_id}"
        listAdvertisedIpContentsbody = {
```

```
"192.168.245.246": "192.168.245.247",
"192.168.197.36": "192.168.197.38",
"192.168.190.11": "192.168.190.11"
}
request.body = UpdateInstanceCrossVpcIpReq(
    advertised_ip_contents=listAdvertisedIpContentsbody
)
response = client.update_instance_cross_vpc_ip(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Modifying the private IP address for cross-VPC access.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceCrossVpcIpRequest{
        request.InstanceId = "{instance_id}"
    }
    var listAdvertisedIpContentsbody = map[string]string{
        "192.168.245.246": "192.168.245.247",
        "192.168.197.36": "192.168.197.38",
        "192.168.190.11": "192.168.190.11",
    }
    request.Body = &model.UpdateInstanceCrossVpcIpReq{
        AdvertisedIpContents: listAdvertisedIpContentsbody,
    }
    response, err := client.UpdateInstanceCrossVpcIp(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The private IP address for cross-VPC access is modified successfully.

Error Codes

See [Error Codes](#).

5.2.6 Querying Kafka Cluster Metadata

Function

This API is used to query Kafka cluster metadata.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/cluster

Table 5-47 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-48 Response body parameters

Parameter	Type	Description
cluster	cluster object	Cluster basic information.

Table 5-49 cluster

Parameter	Type	Description
controller	String	Controller ID.
brokers	Array of brokers objects	Broker list.
topics_count	Integer	Number of topics.
partitions_count	Integer	Number of partitions.
online_partitions_count	Integer	Number of online partitions.
replicas_count	Integer	Number of replicas.
isr_replicas_count	Integer	Total number of in-sync replicas (ISRs).
consumers_count	Integer	Number of consumer groups.

Table 5-50 brokers

Parameter	Type	Description
host	String	Broker IP address.
port	Integer	Port number.
broker_id	String	Broker ID.
is_controller	Boolean	Whether the broker is a controller.
version	String	Server version.
register_time	Long	Broker registration time, which is a Unix timestamp.
is_health	Boolean	Whether Kafka brokers can be connected.

Example Requests

GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/cluster

Example Responses

Status code: 200

Kafka cluster metadata is queried successfully.

```
{
  "cluster": {
    "controller": "2",
    "brokers": [ {
      "host": "192.168.0.159",
      "port": 9093,
      "broker_id": "0",
      "is_controller": false,
      "version": "1.1.0",
      "register_time": 1588754647872,
      "is_health": true
    }, {
      "host": "192.168.0.48",
      "port": 9093,
      "broker_id": "1",
      "is_controller": false,
      "version": "1.1.0",
      "register_time": 1588754647653,
      "is_health": true
    }, {
      "host": "192.168.0.212",
      "port": 9093,
      "broker_id": "2",
      "is_controller": true,
      "version": "1.1.0",
      "register_time": 1588754647284,
      "is_health": true
    }
  ],
  "topics_count": 3,
  "partitions_count": 9,
  "online_partitions_count": 9,
  "replicas_count": 27,
  "isr_replicas_count": 27,
  "consumers_count": 0
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowClusterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowClusterRequest request = new ShowClusterRequest();
request.withInstanceId("{instance_id}");
try {
    ShowClusterResponse response = client.showCluster(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowClusterRequest()
        request.instance_id = "{instance_id}"
        response = client.show_cluster(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```



```
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowClusterRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowCluster(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Kafka cluster metadata is queried successfully.

Error Codes

See [Error Codes](#).

5.2.7 Querying Coordinator Details of a Kafka Instance

Function

This API is used to query coordinator details of a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/coordinators

Table 5-51 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-52 Response body parameters

Parameter	Type	Description
coordinators	Array of coordinators objects	List of coordinators of all consumer groups.

Table 5-53 coordinators

Parameter	Type	Description
group_id	String	Consumer group ID.
id	Integer	Broker ID of the coordinator.
host	String	Address of the coordinator.

Parameter	Type	Description
port	Integer	Port number.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/coordinators
```

Example Responses

Status code: 200

Coordinator details of the Kafka instance are queried successfully.

```
{
  "coordinators" : [ {
    "group_id" : "XXXX",
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9091
  }, {
    "group_id" : "XXXX",
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9092
  }, {
    "group_id" : "XXXX",
    "id" : 2,
    "host" : "172.31.1.15",
    "port" : 9092
  }
]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowCoordinatorsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowCoordinatorsRequest request = new ShowCoordinatorsRequest();
request.withInstanceId("{instance_id}");
try {
    ShowCoordinatorsResponse response = client.showCoordinators(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCoordinatorsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_coordinators(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowCoordinatorsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowCoordinators(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Coordinator details of the Kafka instance are queried successfully.

Error Codes

See [Error Codes](#).

5.2.8 Reassigning Replicas of a Topic for a Kafka Instance

Function

This API is used to reassign replicas of a topic for a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/replicas-reassignment

Table 5-54 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name.

Request Parameters

Table 5-55 Request body parameters

Parameter	Mandatory	Type	Description
partitions	No	Array of partitions objects	Assignment of replicas of the partition after the change.

Table 5-56 partitions

Parameter	Mandatory	Type	Description
partition	No	Integer	Partition ID.

Parameter	Mandatory	Type	Description
replicas	No	Array of integers	ID of the broker where the replica is expected to reside. The first integer in the array represents the leader replica broker ID. All partitions must have the same number of replicas. The number of replicas cannot be larger than the number of brokers.

Response Parameters

None

Example Requests

Reassigning topic partitions, with replicas of partition 1 on brokers 1 and 2, and the leader replica on broker 1

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/replicas-reassignment
```

```
{
  "partitions": [ {
    "partition": 1,
    "replicas": [ 1, 2 ]
  }, {
    "partition": 0,
    "replicas": [ 0, 1 ]
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Reassigning topic partitions, with replicas of partition 1 on brokers 1 and 2, and the leader replica on broker 1

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
import java.util.List;
import java.util.ArrayList;

public class UpdateTopicReplicaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTopicReplicaRequest request = new UpdateTopicReplicaRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        ResetReplicaReq body = new ResetReplicaReq();
        List<Integer> listPartitionsReplicas = new ArrayList<>();
        listPartitionsReplicas.add(0);
        listPartitionsReplicas.add(1);
        List<Integer> listPartitionsReplicas1 = new ArrayList<>();
        listPartitionsReplicas1.add(1);
        listPartitionsReplicas1.add(2);
        List<ResetReplicaReqPartitions> listbodyPartitions = new ArrayList<>();
        listbodyPartitions.add(
            new ResetReplicaReqPartitions()
                .withPartition(1)
                .withReplicas(listPartitionsReplicas1)
        );
        listbodyPartitions.add(
            new ResetReplicaReqPartitions()
                .withPartition(0)
                .withReplicas(listPartitionsReplicas)
        );
        body.withPartitions(listbodyPartitions);
        request.withBody(body);
        try {
            UpdateTopicReplicaResponse response = client.updateTopicReplica(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```


Python

Reassigning topic partitions, with replicas of partition 1 on brokers 1 and 2, and the leader replica on broker 1

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTopicReplicaRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        listReplicasPartitions = [
            0,
            1
        ]
        listReplicasPartitions1 = [
            1,
            2
        ]
        listPartitionsbody = [
            ResetReplicaReqPartitions(
                partition=1,
                replicas=listReplicasPartitions1
            ),
            ResetReplicaReqPartitions(
                partition=0,
                replicas=listReplicasPartitions
            )
        ]
        request.body = ResetReplicaReq(
            partitions=listPartitionsbody
        )
        response = client.update_topic_replica(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Reassigning topic partitions, with replicas of partition 1 on brokers 1 and 2, and the leader replica on broker 1

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTopicReplicaRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    var listReplicasPartitions = []int32{
        int32(0),
        int32(1),
    }
    var listReplicasPartitions1 = []int32{
        int32(1),
        int32(2),
    }
    partitionPartitions:= int32(1)
    partitionPartitions1:= int32(0)
    var listPartitionsbody = []model.ResetReplicaReqPartitions{
        {
            Partition: &partitionPartitions,
            Replicas: &listReplicasPartitions1,
        },
        {
            Partition: &partitionPartitions1,
            Replicas: &listReplicasPartitions,
        },
    }
    request.Body = &model.ResetReplicaReq{
        Partitions: &listPartitionsbody,
    }
    response, err := client.UpdateTopicReplica(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The replicas are reassigned successfully.

Error Codes

See [Error Codes](#).

5.2.9 Modifying Kafka Access Modes

Function

This API is used to modify private or public access for Kafka.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/{engine}/instances/{instance_id}/plain-ssl-switch

Table 5-57 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
engine	Yes	String	Message engine.
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-58 Request body parameters

Parameter	Mandatory	Type	Description
protocol	No	String	Access modes to be enabled or disabled.

Parameter	Mandatory	Type	Description
enable	No	Boolean	<ul style="list-style-type: none"> • true: Enable • false: Disable
user_name	No	String	Username is required when SASL is enabled for the first time. After an instance is created, disabling SASL does not delete the created user. Enabling SASL again does not require the username and an entered username will be invalid.
pass_word	No	String	Password is required when SASL is enabled for the first time.
sasl_enabled_mechanisms	No	Array of strings	Authentication mechanism used after SASL is enabled. Validated only when SASL is enabled for the first time. Reuses are invalid. <ul style="list-style-type: none"> • PLAIN: simple username and password verification. • SCRAM-SHA-512: user credential verification, which is more secure than PLAIN.

Response Parameters

Status code: 200

Table 5-59 Response body parameters

Parameter	Type	Description
job_id	String	Background task ID.
protocol	String	Enabling or disabling the Kafka access mode.
enable	Boolean	Enablement or disablement.

Example Requests

Requiring the username and password when SASL_SSL is enabled for the first time

```
POST https://{endpoint}/v2/{project_id}/{engine}/instances/{instance_id}/plain-ssl-switch
```

```
{
  "protocol": "private_sasl_ssl_enable",
  "enable": true,
  "user_name": "root",
  "pass_word": "password",
  "sasl_enabled_mechanisms": [ "SCRAM-SHA-512", "PLAIN" ]
}
```

Example Responses

None

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.2.10 Querying the Disk Usage Status of Topics

Function

This API is used to query the broker disk usage of topics.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/topics/diskusage

Table 5-60 Path Parameters

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

Table 5-61 Query Parameters

Parameter	Mandatory	Type	Description
minSize	No	String	Querying partitions by the used disk space. Options: 1 KB, 1 MB and 1 GB. Default value: 1 GB.
top	No	String	Querying partitions by top disk usage.
percentage	No	String	Querying partitions by the percentage of the used disk space.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-62 Response body parameters

Parameter	Type	Description
broker_list	Array of DiskusageEntity objects	Broker list.

Table 5-63 DiskusageEntity

Parameter	Type	Description
broker_name	String	Broker name.
data_disk_size	String	Disk capacity.
data_disk_use	String	Used disk space.
data_disk_free	String	Remaining disk space.
data_disk_use_percentage	String	Message label.
status	String	Message label.
topic_list	Array of DiskusageTopicEntity objects	Disk usage list of the topics.

Table 5-64 DiskusageTopicEntity

Parameter	Type	Description
size	String	Disk usage.
topic_name	String	Topic name.
topic_partition	String	Partition.
percentage	Double	Percentage of used disk space.

Example Requests

Querying the disk usage status of topics

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics/diskusage
```

Example Responses

Status code: 200

The query is successful.

```
{
  "broker_list" : [ {
    "broker_name" : "broker-0",
    "data_disk_size" : "66G",
    "data_disk_use" : "53M",
    "data_disk_free" : "63G",
    "data_disk_use_percentage" : "1",
    "status" : "Success get info",
    "topic_list" : [ {
      "size" : "12K",
      "topic_name" : "topic-test",
      "topic_partition" : "2",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "12K",
      "topic_name" : "_consumer_offsets",
      "topic_partition" : "4",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "12K",
      "topic_name" : "_consumer_offsets",
      "topic_partition" : "3",
      "percentage" : 1.7339533025568183E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "_trace",
      "topic_partition" : "6",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "_trace",
      "topic_partition" : "4",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
      "topic_name" : "_trace",
      "topic_partition" : "2",
      "percentage" : 1.1559688683712121E-5
    }, {
      "size" : "8.0K",
```

```
"topic_name" : "__trace",
"topic_partition" : "0",
"percentage" : 1.1559688683712121E-5
}, {
"size" : "8.0K",
"topic_name" : "topic-test",
"topic_partition" : "0",
"percentage" : 1.1559688683712121E-5
}, {
"size" : "8.0K",
"topic_name" : "topic-1568537362",
"topic_partition" : "2",
"percentage" : 1.1559688683712121E-5
}, {
"size" : "8.0K",
"topic_name" : "__consumer_offsets",
"topic_partition" : "7",
"percentage" : 1.1559688683712121E-5
}
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaTopicPartitionDiskusageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowKafkaTopicPartitionDiskusageRequest request = new
        ShowKafkaTopicPartitionDiskusageRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowKafkaTopicPartitionDiskusageResponse response =
            client.showKafkaTopicPartitionDiskusage(request);
            System.out.println(response.toString());
        }
    }
}
```



```
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKafkaTopicPartitionDiskusageRequest()
        request.instance_id = "{instance_id}"
        response = client.show_kafka_topic_partition_diskusage(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowKafkaTopicPartitionDiskusageRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowKafkaTopicPartitionDiskusage(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.2.11 Initiating Partition Reassigning for a Kafka Instance

Function

This API is used to submit a partition rebalancing task to a Kafka instance or calculate estimated rebalancing time.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/kafka/{project_id}/instances/{instance_id}/reassign

Table 5-65 Path Parameters

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

Request Parameters

Table 5-66 Request body parameters

Parameter	Mandatory	Type	Description
reassignments	Yes	Array of PartitionReassignEntity objects	Reassignment plan.
throttle	No	Integer	Reassignment threshold.
is_schedule	No	Boolean	Whether the task is scheduled. If no, is_schedule and execute_at can be left blank. If yes, is_schedule is true and execute_at must be specified.
execute_at	No	Long	Schedule time. The value is a UNIX timestamp, in ms.
time_estimate	No	Boolean	Set true to perform time estimation tasks and false to perform rebalancing tasks.

Table 5-67 PartitionReassignEntity

Parameter	Mandatory	Type	Description
topic	Yes	String	Topic name.
brokers	No	Array of integers	List of brokers to which partitions are reassigned. This parameter is mandatory in automatic assignment.
replication_factor	No	Integer	Replication factor, which can be specified in automatic assignment.

Parameter	Mandatory	Type	Description
assignment	No	Array of TopicAssignment objects	Manually specified assignment plan. The brokers parameter and this parameter cannot be empty at the same time.

Table 5-68 TopicAssignment

Parameter	Mandatory	Type	Description
partition	No	Integer	Partition number in manual assignment.
partition_brokers	No	Array of integers	List of brokers to be assigned to a partition in manual assignment.

Response Parameters

Status code: 200

Table 5-69 Response body parameters

Parameter	Type	Description
job_id	String	Task ID. Only job_id is returned for a rebalancing task.
reassignment_time	Integer	Estimated time, in seconds. Only reassignment_time is returned for a time estimation task.

Example Requests

POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/reassign

```
{
  "reassignments" : [ {
    "topic" : "topic-1513476102",
    "brokers" : [ 0, 1, 2 ],
    "replication_factor" : 3,
    "assignment" : [ {
      "partition" : 0,
      "partition_brokers" : [ 0, 1, 2 ]
    }, {
      "partition" : 1,
      "partition_brokers" : [ 1, 2, 0 ]
    }, {
      "partition" : 2,
      "partition_brokers" : [ 2, 0, 1 ]
    }
  ]
}, {
```

```
"topic" : "topic-1513558717",
"brokers" : [ 0, 1, 4 ],
"replication_factor" : 3,
"assignment" : [ {
  "partition" : 0,
  "partition_brokers" : [ 0, 1, 2 ]
}, {
  "partition" : 1,
  "partition_brokers" : [ 1, 2, 0 ]
}, {
  "partition" : 2,
  "partition_brokers" : [ 2, 0, 1 ]
} ]
},
"throttle" : 10000000,
"time_estimate" : false
}
```

Example Responses

Status code: 200

Rebalancing task submitted or the estimated time is returned.

```
{
  "job_id" : "8a2c259182ab0e9d0182ab1882560009",
  "reassignment_time" : 10
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateReassignmentTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
```

```
.withCredential(auth)
.withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
.build();
CreateReassignmentTaskRequest request = new CreateReassignmentTaskRequest();
request.withInstanceId("{instance_id}");
PartitionReassignRequest body = new PartitionReassignRequest();
List<Integer> listAssignmentPartitionBrokers = new ArrayList<>();
listAssignmentPartitionBrokers.add(2);
listAssignmentPartitionBrokers.add(0);
listAssignmentPartitionBrokers.add(1);
List<Integer> listAssignmentPartitionBrokers1 = new ArrayList<>();
listAssignmentPartitionBrokers1.add(1);
listAssignmentPartitionBrokers1.add(2);
listAssignmentPartitionBrokers1.add(0);
List<Integer> listAssignmentPartitionBrokers2 = new ArrayList<>();
listAssignmentPartitionBrokers2.add(0);
listAssignmentPartitionBrokers2.add(1);
listAssignmentPartitionBrokers2.add(2);
List<TopicAssignment> listReassignmentsAssignment = new ArrayList<>();
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(0)
        .withPartitionBrokers(listAssignmentPartitionBrokers2)
);
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(1)
        .withPartitionBrokers(listAssignmentPartitionBrokers1)
);
listReassignmentsAssignment.add(
    new TopicAssignment()
        .withPartition(2)
        .withPartitionBrokers(listAssignmentPartitionBrokers)
);
List<Integer> listReassignmentsBrokers = new ArrayList<>();
listReassignmentsBrokers.add(0);
listReassignmentsBrokers.add(1);
listReassignmentsBrokers.add(4);
List<Integer> listAssignmentPartitionBrokers3 = new ArrayList<>();
listAssignmentPartitionBrokers3.add(2);
listAssignmentPartitionBrokers3.add(0);
listAssignmentPartitionBrokers3.add(1);
List<Integer> listAssignmentPartitionBrokers4 = new ArrayList<>();
listAssignmentPartitionBrokers4.add(1);
listAssignmentPartitionBrokers4.add(2);
listAssignmentPartitionBrokers4.add(0);
List<Integer> listAssignmentPartitionBrokers5 = new ArrayList<>();
listAssignmentPartitionBrokers5.add(0);
listAssignmentPartitionBrokers5.add(1);
listAssignmentPartitionBrokers5.add(2);
List<TopicAssignment> listReassignmentsAssignment1 = new ArrayList<>();
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(0)
        .withPartitionBrokers(listAssignmentPartitionBrokers5)
);
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(1)
        .withPartitionBrokers(listAssignmentPartitionBrokers4)
);
listReassignmentsAssignment1.add(
    new TopicAssignment()
        .withPartition(2)
        .withPartitionBrokers(listAssignmentPartitionBrokers3)
);
List<Integer> listReassignmentsBrokers1 = new ArrayList<>();
listReassignmentsBrokers1.add(0);
listReassignmentsBrokers1.add(1);
```

```
listReassignmentsBrokers1.add(2);
List<PartitionReassignEntity> listbodyReassignments = new ArrayList<>();
listbodyReassignments.add(
    new PartitionReassignEntity()
        .withTopic("topic-1513476102")
        .withBrokers(listReassignmentsBrokers1)
        .withReplicationFactor(3)
        .withAssignment(listReassignmentsAssignment1)
);
listbodyReassignments.add(
    new PartitionReassignEntity()
        .withTopic("topic-1513558717")
        .withBrokers(listReassignmentsBrokers)
        .withReplicationFactor(3)
        .withAssignment(listReassignmentsAssignment)
);
body.withTimeEstimate(false);
body.withThrottle(10000000);
body.withReassignments(listbodyReassignments);
request.withBody(body);
try {
    CreateReassignmentTaskResponse response = client.createReassignmentTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateReassignmentTaskRequest()
        request.instance_id = "{instance_id}"
        listPartitionBrokersAssignment = [
            2,
```

```

    0,
    1
  ]
  listPartitionBrokersAssignment1 = [
    1,
    2,
    0
  ]
  listPartitionBrokersAssignment2 = [
    0,
    1,
    2
  ]
  listAssignmentReassignments = [
    TopicAssignment(
      partition=0,
      partition_brokers=listPartitionBrokersAssignment2
    ),
    TopicAssignment(
      partition=1,
      partition_brokers=listPartitionBrokersAssignment1
    ),
    TopicAssignment(
      partition=2,
      partition_brokers=listPartitionBrokersAssignment
    )
  ]
  listBrokersReassignments = [
    0,
    1,
    4
  ]
  listPartitionBrokersAssignment3 = [
    2,
    0,
    1
  ]
  listPartitionBrokersAssignment4 = [
    1,
    2,
    0
  ]
  listPartitionBrokersAssignment5 = [
    0,
    1,
    2
  ]
  listAssignmentReassignments1 = [
    TopicAssignment(
      partition=0,
      partition_brokers=listPartitionBrokersAssignment5
    ),
    TopicAssignment(
      partition=1,
      partition_brokers=listPartitionBrokersAssignment4
    ),
    TopicAssignment(
      partition=2,
      partition_brokers=listPartitionBrokersAssignment3
    )
  ]
  listBrokersReassignments1 = [
    0,
    1,
    2
  ]
  listReassignmentsbody = [
    PartitionReassignEntity(
      topic="topic-1513476102",

```



```
        brokers=listBrokersReassignments1,
        replication_factor=3,
        assignment=listAssignmentReassignments1
    ),
    PartitionReassignEntity(
        topic="topic-1513558717",
        brokers=listBrokersReassignments,
        replication_factor=3,
        assignment=listAssignmentReassignments
    )
]
request.body = PartitionReassignRequest(
    time_estimate=False,
    throttle=10000000,
    reassignments=listReassignmentsbody
)
response = client.create_reassignment_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateReassignmentTaskRequest{}
    request.InstanceId = "{instance_id}"
    var listPartitionBrokersAssignment = []int32{
        int32(2),
        int32(0),
        int32(1),
    }
    var listPartitionBrokersAssignment1 = []int32{
        int32(1),
        int32(2),
        int32(0),
    }
```

```
}
var listPartitionBrokersAssignment2 = []int32{
    int32(0),
    int32(1),
    int32(2),
}
partitionAssignment:= int32(0)
partitionAssignment1:= int32(1)
partitionAssignment2:= int32(2)
var listAssignmentReassignments = []model.TopicAssignment{
    {
        Partition: &partitionAssignment,
        PartitionBrokers: &listPartitionBrokersAssignment2,
    },
    {
        Partition: &partitionAssignment1,
        PartitionBrokers: &listPartitionBrokersAssignment1,
    },
    {
        Partition: &partitionAssignment2,
        PartitionBrokers: &listPartitionBrokersAssignment,
    },
}
var listBrokersReassignments = []int32{
    int32(0),
    int32(1),
    int32(4),
}
var listPartitionBrokersAssignment3 = []int32{
    int32(2),
    int32(0),
    int32(1),
}
var listPartitionBrokersAssignment4 = []int32{
    int32(1),
    int32(2),
    int32(0),
}
var listPartitionBrokersAssignment5 = []int32{
    int32(0),
    int32(1),
    int32(2),
}
partitionAssignment3:= int32(0)
partitionAssignment4:= int32(1)
partitionAssignment5:= int32(2)
var listAssignmentReassignments1 = []model.TopicAssignment{
    {
        Partition: &partitionAssignment3,
        PartitionBrokers: &listPartitionBrokersAssignment5,
    },
    {
        Partition: &partitionAssignment4,
        PartitionBrokers: &listPartitionBrokersAssignment4,
    },
    {
        Partition: &partitionAssignment5,
        PartitionBrokers: &listPartitionBrokersAssignment3,
    },
}
var listBrokersReassignments1 = []int32{
    int32(0),
    int32(1),
    int32(2),
}
replicationFactorReassignments:= int32(3)
replicationFactorReassignments1:= int32(3)
var listReassignmentsbody = []model.PartitionReassignEntity{
    {
```

```

Topic: "topic-1513476102",
Brokers: &listBrokersReassignments1,
ReplicationFactor: &replicationFactorReassignments,
Assignment: &listAssignmentReassignments1,
},
{
Topic: "topic-1513558717",
Brokers: &listBrokersReassignments,
ReplicationFactor: &replicationFactorReassignments1,
Assignment: &listAssignmentReassignments,
},
}
timeEstimatePartitionReassignRequest:= false
throttlePartitionReassignRequest:= int32(10000000)
request.Body = &model.PartitionReassignRequest{
TimeEstimate: &timeEstimatePartitionReassignRequest,
Throttle: &throttlePartitionReassignRequest,
Reassignments: listReassignmentsbody,
}
response, err := client.CreateReassignmentTask(request)
if err == nil {
fmt.Printf("%+v\n", response)
} else {
fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Rebalancing task submitted or the estimated time is returned.

Error Codes

See [Error Codes](#).

5.2.12 Disabling Kafka Manager

Function

This API is used to disable Kafka Manager. After Kafka Manager is disabled, the related management APIs will become unavailable.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/kafka/instances/{instance_id}/management

Table 5-70 Path Parameters

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

Request Parameters

None

Response Parameters

None

Example Requests

```
DELETE https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/management
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CloseKafkaManagerSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
```

```
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    CloseKafkaManagerRequest request = new CloseKafkaManagerRequest();
    request.withInstanceId("{instance_id}");
    try {
        CloseKafkaManagerResponse response = client.closeKafkaManager(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CloseKafkaManagerRequest()
        request.instance_id = "{instance_id}"
        response = client.close_kafka_manager(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CloseKafkaManagerRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.CloseKafkaManager(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful.

Error Codes

See [Error Codes](#).

5.2.13 Deleting a User or Client Quota

Function

This API is used to submit a task of deleting a user or client quota from a Kafka instance. A `job_id` is returned if the task succeeds.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Table 5-71 Path Parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .

Request Parameters

Table 5-72 Request body parameters

Parameter	Mandatory	Type	Description
user	No	String	Username. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
client	No	String	Client ID. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
user-default	No	Boolean	Indicates whether to use the default user settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all users. In this case, the username cannot be set. • No: Apply the quota to specific users. In this case, set the username.

Parameter	Mandatory	Type	Description
client-default	No	Boolean	<p>Indicates whether to use the default client settings.</p> <ul style="list-style-type: none"> • Yes: Apply the quota to all clients. In this case, the client ID cannot be set. • No: Apply the quota to specific clients. In this case, set the client ID.

Response Parameters

Status code: 200

Table 5-73 Response body parameters

Parameter	Type	Description
job_id	String	ID of the task for deleting quotas.

Example Requests

Deleting a user or client quota

```
DELETE https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota
```

```
{
  "user" : "",
  "client" : "",
  "user-default" : false,
  "client-default" : true
}
```

Example Responses

Status code: 200

User or client quotas deleted.

```
{
  "job_id" : "ff8080828bdc0f64018bdcadfd8f00d7"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Deleting a user or client quota

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteKafkaUserClientQuotaTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        DeleteKafkaUserClientQuotaTaskRequest request = new DeleteKafkaUserClientQuotaTaskRequest();
        request.withInstanceId("{instance_id}");
        DeleteKafkaUserClientQuotaTaskReq body = new DeleteKafkaUserClientQuotaTaskReq();
        body.withClientDefault(true);
        body.withUserDefault(false);
        body.withClient("");
        body.withUser("");
        request.withBody(body);
        try {
            DeleteKafkaUserClientQuotaTaskResponse response =
client.deleteKafkaUserClientQuotaTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Deleting a user or client quota

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteKafkaUserClientQuotaTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = DeleteKafkaUserClientQuotaTaskReq(
            client_default=True,
            user_default=False,
            client="",
            user=""
        )
        response = client.delete_kafka_user_client_quota_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Deleting a user or client quota

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.DeleteKafkaUserClientQuotaTaskRequest{}
request.InstanceId = "{instance_id}"
clientDefaultDeleteKafkaUserClientQuotaTaskReq:= true
userDefaultDeleteKafkaUserClientQuotaTaskReq:= false
clientDeleteKafkaUserClientQuotaTaskReq:= ""
userDeleteKafkaUserClientQuotaTaskReq:= ""
request.Body = &model.DeleteKafkaUserClientQuotaTaskReq{
    ClientDefault: &clientDefaultDeleteKafkaUserClientQuotaTaskReq,
    UserDefault: &userDefaultDeleteKafkaUserClientQuotaTaskReq,
    Client: &clientDeleteKafkaUserClientQuotaTaskReq,
    User: &userDeleteKafkaUserClientQuotaTaskReq,
}
response, err := client.DeleteKafkaUserClientQuotaTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	User or client quotas deleted.

Error Codes

See [Error Codes](#).

5.2.14 Querying User or Client Quotas

Function

This API is used to query quotas of a Kafka instance. A list of quotas is returned if the query succeeds.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Table 5-74 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 5-75 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset. The records after this offset will be queried.
limit	No	Integer	Maximum number of records that can be returned.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-76 Response body parameters

Parameter	Type	Description
quotas	Array of Quota objects	Client quotas.
count	Integer	Number of user or client quotas.

Table 5-77 Quota

Parameter	Type	Description
user	String	Username. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
client	String	Client ID. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.

Parameter	Type	Description
user-default	Boolean	Indicates whether to use the default user settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all users. In this case, the username cannot be set. • No: Apply the quota to specific users. In this case, set the username.
client-default	Boolean	Indicates whether to use the default client settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all clients. In this case, the client ID cannot be set. • No: Apply the quota to specific clients. In this case, set the client ID.
producer-byte-rate	Long	Production rate limit. The unit is byte/s.
consumer-byte-rate	Long	Consumption rate limit. The unit is byte/s. NOTE Either the production rate limit or the consumption rate limit must be specified.

Example Requests

None

Example Responses

Status code: 200

Client quotas queried.

```
{
  "quotas" : [ {
    "user" : "",
    "client" : "",
    "user-default" : false,
    "client-default" : true,
    "producer-byte-rate" : 2097152,
    "consumer-byte-rate" : 2097152
  } ],
  "count" : 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaUserClientQuotaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowKafkaUserClientQuotaRequest request = new ShowKafkaUserClientQuotaRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowKafkaUserClientQuotaResponse response = client.showKafkaUserClientQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowKafkaUserClientQuotaRequest()
    request.instance_id = "{instance_id}"
    response = client.show_kafka_user_client_quota(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaUserClientQuotaRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowKafkaUserClientQuota(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Client quotas queried.

Error Codes

See [Error Codes](#).

5.2.15 Creating User or Client Quotas

Function

This API is used to submit a task to create user or client quotas for a Kafka instance. A `job_id` is returned if the task succeeds.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Table 5-78 Path Parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .

Request Parameters

Table 5-79 Request body parameters

Parameter	Mandatory	Type	Description
user	No	String	Username. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
client	No	String	Client ID. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
user-default	No	Boolean	Indicates whether to use the default user settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all users. In this case, the username cannot be set. • No: Apply the quota to specific users. In this case, set the username.
client-default	No	Boolean	Indicates whether to use the default client settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all clients. In this case, the client ID cannot be set. • No: Apply the quota to specific clients. In this case, set the client ID.
producer-byte-rate	No	Long	Production rate limit. The unit is byte/s.
consumer-byte-rate	No	Long	Consumption rate limit. The unit is byte/s. NOTE Either the production rate limit or the consumption rate limit must be specified.

Response Parameters

Status code: 200

Table 5-80 Response body parameters

Parameter	Type	Description
job_id	String	ID of the task for creating quotas.

Example Requests

Creating user or client quotas

POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

```
{
  "user" : "",
  "client" : "",
  "user-default" : false,
  "client-default" : true,
  "producer-byte-rate" : 3145728,
  "consumer-byte-rate" : 2097152
}
```

Example Responses

Status code: 200

User or client quotas created.

```
{
  "job_id" : "ff8080828bdc0f64018bdcadfd8f00d7"
}
```

Status Codes

Status Code	Description
200	User or client quotas created.

Error Codes

See [Error Codes](#).

5.2.16 Modifying User or Client Quotas

Function

This API is used to submit a task to modify user or client quotas of a Kafka instance. A job_id is returned if the task succeeds.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota

Table 5-81 Path Parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .

Request Parameters

Table 5-82 Request body parameters

Parameter	Mandatory	Type	Description
user	No	String	Username. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
client	No	String	Client ID. Either the username or the client ID must be specified if the quotas are not applied to all users or clients.
user-default	No	Boolean	Indicates whether to use the default user settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all users. In this case, the username cannot be set. • No: Apply the quota to specific users. In this case, set the username.
client-default	No	Boolean	Indicates whether to use the default client settings. <ul style="list-style-type: none"> • Yes: Apply the quota to all clients. In this case, the client ID cannot be set. • No: Apply the quota to specific clients. In this case, set the client ID.

Parameter	Mandatory	Type	Description
producer-byte-rate	No	Long	Production rate limit. The unit is byte/s.
consumer-byte-rate	No	Long	Consumption rate limit. The unit is byte/s. NOTE Either the production rate limit or the consumption rate limit must be specified.

Response Parameters

Status code: 200

Table 5-83 Response body parameters

Parameter	Type	Description
job_id	String	ID of the task for modifying quotas.

Example Requests

Modifying user or client quotas

```
PUT https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/kafka-user-client-quota
```

```
{
  "user" : "",
  "client" : "",
  "user-default" : false,
  "client-default" : true,
  "producer-byte-rate" : 3145728,
  "consumer-byte-rate" : 2097152
}
```

Example Responses

Status code: 200

User or client quotas modified.

```
{
  "job_id" : "8abfa7b38ba79a20018ba9afc550576a"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying user or client quotas

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateKafkaUserClientQuotaTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateKafkaUserClientQuotaTaskRequest request = new UpdateKafkaUserClientQuotaTaskRequest();
        request.withInstanceId("{instance_id}");
        UpdateKafkaUserClientQuotaTaskReq body = new UpdateKafkaUserClientQuotaTaskReq();
        body.withConsumerByteRate(2097152L);
        body.withProducerByteRate(3145728L);
        body.withClientDefault(true);
        body.withUserDefault(false);
        body.withClient("");
        body.withUser("");
        request.withBody(body);
        try {
            UpdateKafkaUserClientQuotaTaskResponse response =
client.updateKafkaUserClientQuotaTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Modifying user or client quotas

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateKafkaUserClientQuotaTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateKafkaUserClientQuotaTaskReq(
            consumer_byte_rate=2097152,
            producer_byte_rate=3145728,
            client_default=True,
            user_default=False,
            client="",
            user=""
        )
        response = client.update_kafka_user_client_quota_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Modifying user or client quotas

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()
```

```

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateKafkaUserClientQuotaTaskRequest{}
request.InstanceId = "{instance_id}"
consumerByteRateUpdateKafkaUserClientQuotaTaskReq:= int64(2097152)
producerByteRateUpdateKafkaUserClientQuotaTaskReq:= int64(3145728)
clientDefaultUpdateKafkaUserClientQuotaTaskReq:= true
userDefaultUpdateKafkaUserClientQuotaTaskReq:= false
clientUpdateKafkaUserClientQuotaTaskReq:= ""
userUpdateKafkaUserClientQuotaTaskReq:= ""
request.Body = &model.UpdateKafkaUserClientQuotaTaskReq{
    ConsumerByteRate: &consumerByteRateUpdateKafkaUserClientQuotaTaskReq,
    ProducerByteRate: &producerByteRateUpdateKafkaUserClientQuotaTaskReq,
    ClientDefault: &clientDefaultUpdateKafkaUserClientQuotaTaskReq,
    UserDefault: &userDefaultUpdateKafkaUserClientQuotaTaskReq,
    Client: &clientUpdateKafkaUserClientQuotaTaskReq,
    User: &userUpdateKafkaUserClientQuotaTaskReq,
}
response, err := client.UpdateKafkaUserClientQuotaTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	User or client quotas modified.

Error Codes

See [Error Codes](#).

5.3 Smart Connect

5.3.1 Enabling Smart Connect (Pay-per-Use Instance)

Function

This API is used to enable Smart Connect so you can create a connector.

This API is only available for pay-per-use instances.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/connector

Table 5-84 Path Parameters

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

Request Parameters

Table 5-85 Request body parameters

Parameter	Mandatory	Type	Description
specification	No	String	Bandwidth for deploying Smart Connect, that is, the maximum amount of data transferred per unit time. Use the bandwidth of the current instance. This parameter is mandatory only for earlier instance flavors. Value range: <ul style="list-style-type: none">• 100 MB• 300 MB• 600 MB• 1200 MB
node_cnt	No	String	Number of connectors. Min.: 2. The default value is 2 if it is not specified.
spec_code	No	String	Specification code of the connector. This parameter is mandatory only for old instance flavors.

Response Parameters

Status code: 200

Table 5-86 Response body parameters

Parameter	Type	Description
job_id	String	Task ID.
connector_id	String	Instance dump ID.

Example Requests

- To enable Smart Connect for pay-per-use instances using new flavors, set the number of connectors to 2.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector
{
  "node_cnt" : 2
}
```

- To enable Smart Connect for pay-per-use instances using old flavors, set the size of connectors to 100 MB and the number of them to 2.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector
{
  "specification" : "100MB",
  "node_cnt" : 2,
  "spec_code" : "kafka.c3.mini.connector"
}
```

Example Responses

Status code: 200

Successful.

```
{
  "job_id" : "7c3ec20c-11de-4df9-acc0-7ef1dea25dfe",
  "connector_id" : "55b78880-9077-4c74-ad5a-6868555f76a4"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- To enable Smart Connect for pay-per-use instances using new flavors, set the number of connectors to 2.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
```

```
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateConnectorRequest request = new CreateConnectorRequest();
        request.withInstanceId("{instance_id}");
        CreateConnectorReq body = new CreateConnectorReq();
        body.withNodeCnt("2");
        request.withBody(body);
        try {
            CreateConnectorResponse response = client.createConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- To enable Smart Connect for pay-per-use instances using old flavors, set the size of connectors to 100 MB and the number of them to 2.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
```

running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateConnectorRequest request = new CreateConnectorRequest();
request.withInstanceId("{instance_id}");
CreateConnectorReq body = new CreateConnectorReq();
body.withSpecCode("kafka.c3.mini.connector");
body.withNodeCnt("2");
body.withSpecification(CreateConnectorReq.SpecificationEnum.fromValue("100MB"));
request.withBody(body);
try {
    CreateConnectorResponse response = client.createConnector(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- To enable Smart Connect for pay-per-use instances using new flavors, set the number of connectors to 2.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = CreateConnectorRequest()
    request.instance_id = "{instance_id}"
    request.body = CreateConnectorReq(
        node_cnt="2"
    )
    response = client.create_connector(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- To enable Smart Connect for pay-per-use instances using old flavors, set the size of connectors to 100 MB and the number of them to 2.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateConnectorReq(
            spec_code="kafka.c3.mini.connector",
            node_cnt="2",
            specification="100MB"
        )
        response = client.create_connector(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- To enable Smart Connect for pay-per-use instances using new flavors, set the number of connectors to 2.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorRequest{}
    request.InstanceId = "{instance_id}"
    nodeCntCreateConnectorReq := "2"
    request.Body = &model.CreateConnectorReq{
        NodeCnt: &nodeCntCreateConnectorReq,
    }
    response, err := client.CreateConnector(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- To enable Smart Connect for pay-per-use instances using old flavors, set the size of connectors to 100 MB and the number of them to 2.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```

Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateConnectorRequest{
    request.InstanceId = "{instance_id}"
    specCodeCreateConnectorReq:= "kafka.c3.mini.connector"
    nodeCntCreateConnectorReq:= "2"
    specificationCreateConnectorReq:= model.GetCreateConnectorReqSpecificationEnum().E_100_MB
    request.Body = &model.CreateConnectorReq{
        SpecCode: &specCodeCreateConnectorReq,
        NodeCnt: &nodeCntCreateConnectorReq,
        Specification: &specificationCreateConnectorReq,
    }
}
response, err := client.CreateConnector(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful.

Error Codes

See [Error Codes](#).

5.3.2 Disabling Smart Connect (Pay-per-Use Instance)

Function

This API is used to disable Smart Connect for a pay-per-use instance.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/delete-connector

Table 5-87 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-88 Response body parameters

Parameter	Type	Description
job_id	String	ID of the job for asynchronously executing the deletion task.

Example Requests

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/delete-connector
```

Example Responses

Status code: 200

Smart Connect disabling task submitted successfully.

```
{
  "job_id" : "d366178c-29ea-4d5c-a344-fa2ece4a1836"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class DeleteConnectorSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteConnectorRequest request = new DeleteConnectorRequest();
        request.withInstanceId("{instance_id}");
        try {
            DeleteConnectorResponse response = client.deleteConnector(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()
```



```
try:
    request = DeleteConnectorRequest()
    request.instance_id = "{instance_id}"
    response = client.delete_connector(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteConnectorRequest{
        request.InstanceId = "{instance_id}"
    }
    response, err := client.DeleteConnector(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Smart Connect disabling task submitted successfully.

Error Codes

See [Error Codes](#).

5.3.3 Creating a Smart Connect Task

Function

This API is used to create a Smart Connect task.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/connector/tasks

Table 5-89 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-90 Request body parameters

Parameter	Mandatory	Type	Description
task_name	No	String	Smart Connect task name.
start_later	No	Boolean	Indicates whether to start a task later. false : to create a task and start immediately; true : to create a task and manually start it later in the task list.

Parameter	Mandatory	Type	Description
topics	No	String	Topic of a Smart Connect task.
topics_regex	No	String	Regular expression of the topic of a Smart Connect task.
source_type	No	String	Source type of a Smart Connect task.
source_task	No	SmartConnectTaskReqSourceConfig object	Source configuration of a Smart Connect task.
sink_type	No	String	Target type of a Smart Connect task.
sink_task	No	SmartConnectTaskReqSinkConfig object	Target type of a Smart Connect task.

Table 5-91 SmartConnectTaskReqSourceConfig

Parameter	Mandatory	Type	Description
redis_address	No	String	Redis instance address. (Mandatory only when the source type is Redis.)
redis_type	No	String	Redis instance type. (Mandatory only when the source type is Redis.)
dcs_instance_id	No	String	DCS instance ID. (Mandatory only when the source type is Redis.)
redis_password	No	String	Redis password. (Mandatory only when the source type is Redis.)
sync_mode	No	String	Synchronization type: RDB_ONLY indicates full synchronization; CUSTOM_OFFSET indicates full and incremental synchronization. (Mandatory only when the source type is Redis.)

Parameter	Mandatory	Type	Description
full_sync_wait_ms	No	Integer	Interval of full synchronization retries, in ms. (Mandatory only when the source type is Redis.)
full_sync_max_retry	No	Integer	Max. retries of full synchronization. (Mandatory only when the source type is Redis.)
ratelimit	No	Integer	Rate limit, in KB/s. -1: There is no rate limit. (Mandatory only when the source type is Redis.)
current_cluster_name	No	String	Current Kafka instance name. (Mandatory only when the source type is Kafka.)
cluster_name	No	String	Target Kafka instance name. (Mandatory only when the source type is Kafka.)
user_name	No	String	Username set when SASL_SSL was enabled for a target Kafka instance, or when a SASL_SSL user was created. (Mandatory only when the source type is Kafka and the target Kafka authentication mode is SASL_SSL .)
password	No	String	Password set when SASL_SSL was enabled for a target Kafka instance, or when a SASL_SSL user was created. (Mandatory only when the source type is Kafka and the target Kafka authentication mode is SASL_SSL .)
sasl_mechanism	No	String	Target Kafka authentication mode. (Mandatory only when the source type is Kafka and the authentication mode is SASL_SSL .)
instance_id	No	String	Target Kafka instance ID. (Mandatory only when the source type is Kafka. Specify either instance_id or bootstrap_servers .)

Parameter	Mandatory	Type	Description
bootstrap_servers	No	String	Target Kafka instance address. (Mandatory only when the source type is Kafka. Specify either instance_id or bootstrap_servers .)
security_protocol	No	String	Target Kafka authentication. (Mandatory only when the source type is Kafka). There are two authentication modes: <ul style="list-style-type: none"> • SASL_SSL: SASL_SSL is enabled. • PLAINTEXT: SASL_SSL is disabled.
direction	No	String	Sync direction: pull replicates data from the target Kafka instance to the current one; push replicates data from the source Kafka instance to the target one; two-way replicates data from the source and target Kafka instance data to each other. (Mandatory only when the source type is Kafka.)
sync_consumer_offsets_enabled	No	Boolean	Indicates whether to sync the consumption progress. (Mandatory only when the source type is Kafka.)
replication_factor	No	Integer	Number of topic replicas when a topic is automatically created in the peer instance. The value of this parameter cannot exceed the number of brokers in the peer instance. This parameter overrides the default.replication.factor parameter configured in the target instance. (Mandatory only when the source type is Kafka.)

Parameter	Mandatory	Type	Description
task_num	No	Integer	Number of data replication tasks. The default value is 2 . You are advised to use the default value. If the sync direction is set to two-way, the actual number of tasks will be twice the number of tasks you configure here. (Mandatory only when the source type is Kafka.)
rename_topic_enabled	No	Boolean	Indicates whether to rename a topic. If yes, add the alias of the source Kafka instance before the target topic name to form a new name of the target topic. (Mandatory only when the source type is Kafka.)
provenance_header_enabled	No	Boolean	The target topic receives the replicated messages. The message header contains the message source. If you select Both for Sync Direction , enable Add Source Header to prevent infinite replication. (Mandatory only when the source type is Kafka.)
consumer_strategy	No	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Mandatory only when the source type is Kafka.)
compression_type	No	String	Compression algorithm to use for copying messages. (Mandatory only when the source type is Kafka.) <ul style="list-style-type: none"> • none • gzip • snappy • lz4 • zstd

Parameter	Mandatory	Type	Description
topics_mapping	No	String	Topic mapping, which is used to customize the target topic name. Rename Topic and Topic Mapping cannot be configured at the same time. Topic mapping format: <i>source topic.target topic</i> . Use commas (,) to separate multiple topic mappings, for example, topic-sc-1:topic-sc-2,topic-sc-3:topic-sc-4 . (Mandatory only when the source type is Kafka.)

Table 5-92 SmartConnectTaskReqSinkConfig

Parameter	Mandatory	Type	Description
redis_address	No	String	Redis instance address. (Mandatory only when the target type is Redis.)
redis_type	No	String	Redis instance type. (Mandatory only when the target type is Redis.)
dcs_instance_id	No	String	DCS instance ID. (Mandatory only when the target type is Redis.)
redis_password	No	String	Redis password. (Mandatory only when the target type is Redis.)
consumer_strategy	No	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Mandatory only when the target type is OBS.)
destination_file_type	No	String	Dump file format. Only TEXT is supported. (Mandatory only when the target type is OBS.)
deliver_time_interval	No	Integer	Data dumping intervals, in seconds. The default interval is 300s. (Mandatory only when the target type is OBS.)

Parameter	Mandatory	Type	Description
access_key	No	String	AK: access key ID. (Mandatory only when the target type is OBS.)
secret_key	No	String	SK: secret access key used together with the access key ID. (Mandatory only when the target type is OBS.)
obs_bucket_name	No	String	Dumping address, which is the OBS bucket used to store the topic data. (Mandatory only when the target type is OBS.)
obs_path	No	String	Dumping directory, which is the directory for storing topic files dumped to OBS. Use slashes (/) to separate directory levels. (Mandatory only when the target type is OBS.)
partition_format	No	String	Time directory format. (Mandatory only when the target type is OBS.) <ul style="list-style-type: none"> • 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.
record_delimiter	No	String	Line break, which is used to separate the user data that is written into the dump file. (Mandatory only when the target type is OBS.) Value range: <ul style="list-style-type: none"> • Comma (,) • Semicolon (;) • Vertical bar () • Newline (\n) • NULL

Parameter	Mandatory	Type	Description
store_keys	No	Boolean	Specifies whether to dump keys. (Mandatory only when the target type is OBS.)

Response Parameters

Status code: 200

Table 5-93 Response body parameters

Parameter	Type	Description
task_name	String	Smart Connect task name.
topics	String	Topic of a Smart Connect task.
topics_regex	String	Regular expression of the topic of a Smart Connect task.
source_type	String	Source type of a Smart Connect task.
source_task	SmartConnectTaskRespSourceConfig object	Source configuration of a Smart Connect task.
sink_type	String	Target type of a Smart Connect task.
sink_task	SmartConnectTaskRespSinkConfig object	Target type of a Smart Connect task.
id	String	ID of a Smart Connect task.
status	String	Smart Connect task status.
create_time	Long	Time when the Smart Connect task was created.

Table 5-94 SmartConnectTaskRespSourceConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the source type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the source type is Redis.)
dcs_instance_id	String	DCS instance ID. (Displayed only when the source type is Redis.)

Parameter	Type	Description
sync_mode	String	Synchronization type: RDB_ONLY indicates full synchronization; CUSTOM_OFFSET indicates full and incremental synchronization. (Displayed only when the source type is Redis.)
full_sync_wait_ms	Integer	Interval of full synchronization retries, in ms. (Displayed only when the source type is Redis.)
full_sync_max_retry	Integer	Max. retries of full synchronization. (Displayed only when the source type is Redis.)
ratelimit	Integer	Rate limit, in KB/s. -1 : disable. (Displayed only when the source type is Redis.)
current_cluster_name	String	Current Kafka instance name. (Displayed only when the source type is Kafka.)
cluster_name	String	Target Kafka instance name. (Displayed only when the source type is Kafka.)
user_name	String	Username of the target Kafka instance. (Displayed only when the source type is Kafka.)
sasl_mechanism	String	Target Kafka authentication mode. (Displayed only when the source type is Kafka.)
instance_id	String	Target Kafka instance ID. (Displayed only when the source type is Kafka.)
bootstrap_servers	String	Target Kafka instance address. (Displayed only when the source type is Kafka.)
security_protocol	String	Target Kafka authentication. (Displayed only when the source type is Kafka.)
direction	String	Sync direction. (Displayed only when the source type is Kafka.)
sync_consumer_of_fsets_enabled	Boolean	Indicates whether to sync the consumption progress. (Displayed only when the source type is Kafka.)

Parameter	Type	Description
replication_factor	Integer	Number of replicas. (Displayed only when the source type is Kafka.)
task_num	Integer	Number of tasks. (Displayed only when the source type is Kafka.)
rename_topic_enabled	Boolean	Indicates whether to rename a topic. (Displayed only when the source type is Kafka.)
provenance_header_enabled	Boolean	Indicates whether to add the source header. (Displayed only when the source type is Kafka.)
consumer_strategy	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the source type is Kafka.)
compression_type	String	Compression algorithm. (Displayed only when the source type is Kafka.)
topics_mapping	String	Topic mapping. (Displayed only when the source type is Kafka.)

Table 5-95 SmartConnectTaskRespSinkConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the target type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the target type is Redis.)
dcs_instance_id	String	DCS instance ID. (Displayed only when the target type is Redis.)
target_db	Integer	Target database. The default value is -1 . (Displayed only when the target type is Redis.)
consumer_strategy	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the target type is OBS.)
destination_file_type	String	Dump file format. Only TEXT is supported. (Displayed only when the target type is OBS.)

Parameter	Type	Description
deliver_time_interval	Integer	Dumping period (s). (Displayed only when the target type is OBS.)
obs_bucket_name	String	Dumping address. (Displayed only when the target type is OBS.)
obs_path	String	Dump directory. (Displayed only when the target type is OBS.)
partition_format	String	Time directory format. (Displayed only when the target type is OBS.)
record_delimiter	String	Line break. (Displayed only when the target type is OBS.)
store_keys	Boolean	Storage key. (Displayed only when the target type is OBS.)
obs_part_size	Integer	Size (in bytes) of each file to be uploaded. The default value is 5242880 . (Displayed only when the target type is OBS.)
flush_size	Integer	flush_size. (Displayed only when the target type is OBS.)
timezone	String	Time zone. (Displayed only when the target type is OBS.)
schema_generator_class	String	schema_generator class. The default value is io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Displayed only when the target type is OBS.)
partitioner_class	String	partitioner class. The default value is io.confluent.connect.storage.partitionner.TimeBasedPartitionner . (Displayed only when the target type is OBS.)
value_converter	String	value_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)
key_converter	String	key_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)
kv_delimiter	String	kv_delimiter. The default value is : . (Displayed only when the target type is OBS.)

Example Requests

- Creating a dumping task that starts immediately.

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks

```
{
  "task_name": "smart-connect-1",
  "start_later": false,
  "source_type": "NONE",
  "topics_regex": "topic-obs*",
  "sink_type": "OBS_SINK",
  "sink_task": {
    "consumer_strategy": "earliest",
    "destination_file_type": "TEXT",
    "deliver_time_interval": 300,
    "access_key": "*****",
    "secret_key": "*****",
    "obs_bucket_name": "obs_bucket",
    "obs_path": "obsTransfer-1810125534",
    "partition_format": "yyyy/MM/dd/HH/mm",
    "record_delimiter": "\\n",
    "store_keys": false
  }
}
```

- Creating a Kafka data replication task that starts later.

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks

```
{
  "task_name": "smart-connect-2",
  "start_later": true,
  "source_type": "KAFKA_REPLICATOR_SOURCE",
  "source_task": {
    "current_cluster_name": "A",
    "cluster_name": "B",
    "user_name": "user1",
    "password": "*****",
    "saslm_mechanism": "SCRAM-SHA-512",
    "instance_id": "b54c9dd8-*****-*****",
    "direction": "two-way",
    "sync_consumer_offsets_enabled": false,
    "replication_factor": 3,
    "task_num": 2,
    "rename_topic_enabled": false,
    "provenance_header_enabled": true,
    "consumer_strategy": "latest",
    "compression_type": "snappy",
    "topics_mapping": "topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
  }
}
```

- Creating a Redis data replication task that starts immediately: full synchronization, 10 maximum retry times, 10,000 ms retry interval, and 10 KB/s bandwidth limit.

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks

```
{
  "task_name": "smart-connect-3",
  "start_later": false,
  "source_type": "REDIS_REPLICATOR_SOURCE",
  "source_task": {
    "redis_address": "192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379",
    "redis_type": "cluster",
    "redis_password": "*****",
    "sync_mode": "RDB_ONLY",
  }
}
```

```
"full_sync_max_retry" : 10,
"full_sync_wait_ms" : 10000,
"ratelimit" : 10
},
"topics" : "topic-sc-3",
"sink_type" : "REDIS_REPLICATOR_SINK",
"sink_task" : {
  "redis_address" : "192.168.119.51:6379",
  "redis_type" : "standalone",
  "redis_password" : "*****"
}
}
```

Example Responses

Status code: 200

Successful.

```
{
  "task_name" : "smart-connect-121248117",
  "topics" : "topic-sc",
  "source_task" : {
    "redis_address" : "192.168.91.179:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3cae1e7cd",
    "sync_mode" : "RDB_ONLY",
    "full_sync_wait_ms" : 13000,
    "full_sync_max_retry" : 4,
    "ratelimit" : -1
  },
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",
    "target_db" : -1
  },
  "sink_type" : "REDIS_REPLICATOR_SINK",
  "id" : "8a205bbd-7181-4b5e-9bd6-37274ce84577",
  "status" : "RUNNING",
  "create_time" : 1708427753133
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Creating a dumping task that starts immediately.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorTaskSolution {

  public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();
request.withInstanceId("{instance_id}");
CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();
SmartConnectTaskReqSinkConfig sinkTaskbody = new SmartConnectTaskReqSinkConfig();
sinkTaskbody.withConsumerStrategy("earliest")
    .withDestinationFileType("TEXT")
    .withDeliverTimeInterval(300)
    .withAccessKey("*****")
    .withSecretKey("*****")
    .withObsBucketName("obs_bucket")
    .withObsPath("obsTransfer-1810125534")
    .withPartitionFormat("yyyy/MM/dd/HH/mm")
    .withRecordDelimiter("\n")
    .withStoreKeys(false);
body.withSinkTask(sinkTaskbody);
body.withSinkType(CreateSmartConnectTaskReq.SinkTypeEnum.fromValue("OBS_SINK"));
body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("NONE"));
body.withTopicsRegex("topic-obs*");
body.withStartLater(false);
body.withTaskName("smart-connect-1");
request.withBody(body);
try {
    CreateConnectorTaskResponse response = client.createConnectorTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **Creating a Kafka data replication task that starts later.**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class CreateConnectorTaskSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR_REGION>"))  
            .build();  
        CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();  
        request.withInstanceId("{instance_id}");  
        CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();  
        SmartConnectTaskReqSourceConfig sourceTaskbody = new SmartConnectTaskReqSourceConfig();  
        sourceTaskbody.withCurrentClusterName("A")  
            .withClusterName("B")  
            .withUserName("user1")  
            .withPassword("*****")  
            .withSaslMechanism("SCRAM-SHA-512")  
            .withInstanceId("b54c9dd8-*****-*****")  
            .withDirection("two-way")  
            .withSyncConsumerOffsetsEnabled(false)  
            .withReplicationFactor(3)  
            .withTaskNum(2)  
            .withRenameTopicEnabled(false)  
            .withProvenanceHeaderEnabled(true)  
            .withConsumerStrategy("latest")  
            .withCompressionType("snappy")  
            .withTopicsMapping("topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4");  
        body.withSourceTask(sourceTaskbody);  
  
        body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("KAFKA_REPLICATOR_  
SOURCE"));  
        body.withStartLater(true);  
        body.withTaskName("smart-connect-2");  
        request.withBody(body);  
        try {  
            CreateConnectorTaskResponse response = client.createConnectorTask(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

- Creating a Redis data replication task that starts immediately: full synchronization, 10 maximum retry times, 10,000 ms retry interval, and 10 KB/s bandwidth limit.


```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateConnectorTaskRequest request = new CreateConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        CreateSmartConnectTaskReq body = new CreateSmartConnectTaskReq();
        SmartConnectTaskReqSinkConfig sinkTaskbody = new SmartConnectTaskReqSinkConfig();
        sinkTaskbody.withRedisAddress("192.168.119.51:6379")
            .withRedisType("standalone")
            .withRedisPassword("*****");
        SmartConnectTaskReqSourceConfig sourceTaskbody = new SmartConnectTaskReqSourceConfig();
        sourceTaskbody.withRedisAddress("192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379")
            .withRedisType("cluster")
            .withRedisPassword("*****")
            .withSyncMode("RDB_ONLY")
            .withFullSyncWaitMs(10000)
            .withFullSyncMaxRetry(10)
            .withRatelimit(10);
        body.withSinkTask(sinkTaskbody);

        body.withSinkType(CreateSmartConnectTaskReq.SinkTypeEnum.fromValue("REDIS_REPLICATOR_SINK"));
        body.withSourceTask(sourceTaskbody);

        body.withSourceType(CreateSmartConnectTaskReq.SourceTypeEnum.fromValue("REDIS_REPLICATOR_S
        OURCE"));
        body.withTopics("topic-sc-3");
        body.withStartLater(false);
        body.withTaskName("smart-connect-3");
        request.withBody(body);
        try {
            CreateConnectorTaskResponse response = client.createConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

- Creating a dumping task that starts immediately.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        sinkTaskbody = SmartConnectTaskReqSinkConfig(
            consumer_strategy="earliest",
            destination_file_type="TEXT",
            deliver_time_interval=300,
            access_key="*****",
            secret_key="*****",
            obs_bucket_name="obs_bucket",
            obs_path="obsTransfer-1810125534",
            partition_format="yyyy/MM/dd/HH/mm",
            record_delimiter="\n",
            store_keys=False
        )
        request.body = CreateSmartConnectTaskReq(
            sink_task=sinkTaskbody,
            sink_type="OBS_SINK",
            source_type="NONE",
            topics_regex="topic-obs*",
            start_later=False,
            task_name="smart-connect-1"
        )
        response = client.create_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Creating a Kafka data replication task that starts later.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        sourceTaskbody = SmartConnectTaskReqSourceConfig(
            current_cluster_name="A",
            cluster_name="B",
            user_name="user1",
            password="*****",
            sasl_mechanism="SCRAM-SHA-512",
            instance_id="b54c9dd8-*****-*****",
            direction="two-way",
            sync_consumer_offsets_enabled=False,
            replication_factor=3,
            task_num=2,
            rename_topic_enabled=False,
            provenance_header_enabled=True,
            consumer_strategy="latest",
            compression_type="snappy",
            topics_mapping="topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
        )
        request.body = CreateSmartConnectTaskReq(
            source_task=sourceTaskbody,
            source_type="KAFKA_REPLICATOR_SOURCE",
            start_later=True,
            task_name="smart-connect-2"
        )
        response = client.create_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Creating a Redis data replication task that starts immediately: full synchronization, 10 maximum retry times, 10,000 ms retry interval, and 10 KB/s bandwidth limit.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        sinkTaskbody = SmartConnectTaskReqSinkConfig(
            redis_address="192.168.119.51:6379",
            redis_type="standalone",
            redis_password="*****"
        )
        sourceTaskbody = SmartConnectTaskReqSourceConfig(
            redis_address="192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379",
            redis_type="cluster",
            redis_password="*****",
            sync_mode="RDB_ONLY",
            full_sync_wait_ms=10000,
            full_sync_max_retry=10,
            ratelimit=10
        )
        request.body = CreateSmartConnectTaskReq(
            sink_task=sinkTaskbody,
            sink_type="REDIS_REPLICATOR_SINK",
            source_task=sourceTaskbody,
            source_type="REDIS_REPLICATOR_SOURCE",
            topics="topic-sc-3",
            start_later=False,
            task_name="smart-connect-3"
        )
        response = client.create_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- Creating a dumping task that starts immediately.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorTaskRequest{
        request.InstanceId = "{instance_id}"
        consumerStrategySinkTask:= "earliest"
        destinationFileTypeSinkTask:= "TEXT"
        deliverTimeIntervalSinkTask:= int32(300)
        accessKeySinkTask:= "*****"
        secretKeySinkTask:= "*****"
        obsBucketNameSinkTask:= "obs_bucket"
        obsPathSinkTask:= "obsTransfer-1810125534"
        partitionFormatSinkTask:= "yyyy/MM/dd/HH/mm"
        recordDelimiterSinkTask:= "\n"
        storeKeysSinkTask:= false
        sinkTaskbody := &model.SmartConnectTaskReqSinkConfig{
            ConsumerStrategy: &consumerStrategySinkTask,
            DestinationFileType: &destinationFileTypeSinkTask,
            DeliverTimeInterval: &deliverTimeIntervalSinkTask,
            AccessKey: &accessKeySinkTask,
            SecretKey: &secretKeySinkTask,
            ObsBucketName: &obsBucketNameSinkTask,
            ObsPath: &obsPathSinkTask,
            PartitionFormat: &partitionFormatSinkTask,
            RecordDelimiter: &recordDelimiterSinkTask,
            StoreKeys: &storeKeysSinkTask,
        }
        sinkTypeCreateSmartConnectTaskReq:=
        model.GetCreateSmartConnectTaskReqSinkTypeEnum().OBS_SINK
        sourceTypeCreateSmartConnectTaskReq:=
        model.GetCreateSmartConnectTaskReqSourceTypeEnum().NONE
        topicsRegexCreateSmartConnectTaskReq:= "topic-obs*"
        startLaterCreateSmartConnectTaskReq:= false
        taskNameCreateSmartConnectTaskReq:= "smart-connect-1"
        request.Body = &model.CreateSmartConnectTaskReq{
            SinkTask: sinkTaskbody,
            SinkType: &sinkTypeCreateSmartConnectTaskReq,
            SourceType: &sourceTypeCreateSmartConnectTaskReq,
            TopicsRegex: &topicsRegexCreateSmartConnectTaskReq,
            StartLater: &startLaterCreateSmartConnectTaskReq,
            TaskName: &taskNameCreateSmartConnectTaskReq,
        }
    }
    response, err := client.CreateConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Creating a Kafka data replication task that starts later.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    currentClusterNameSourceTask:= "A"
    clusterNameSourceTask:= "B"
    userNameSourceTask:= "user1"
    passwordSourceTask:= "*****"
    saslMechanismSourceTask:= "SCRAM-SHA-512"
    instanceIdSourceTask:= "b54c9dd8_*****_*****"
    directionSourceTask:= "two-way"
    syncConsumerOffsetsEnabledSourceTask:= false
    replicationFactorSourceTask:= int32(3)
    taskNumSourceTask:= int32(2)
    renameTopicEnabledSourceTask:= false
    provenanceHeaderEnabledSourceTask:= true
    consumerStrategySourceTask:= "latest"
    compressionTypeSourceTask:= "snappy"
    topicsMappingSourceTask:= "topic-sc-1:topic-sc-3,topic-sc-2:topic-sc-4"
    sourceTaskbody := &model.SmartConnectTaskReqSourceConfig{
        CurrentClusterName: &currentClusterNameSourceTask,
        ClusterName: &clusterNameSourceTask,
        UserName: &userNameSourceTask,
        Password: &passwordSourceTask,
        SaslMechanism: &saslMechanismSourceTask,
        InstanceId: &instanceIdSourceTask,
        Direction: &directionSourceTask,
        SyncConsumerOffsetsEnabled: &syncConsumerOffsetsEnabledSourceTask,
        ReplicationFactor: &replicationFactorSourceTask,
        TaskNum: &taskNumSourceTask,
        RenameTopicEnabled: &renameTopicEnabledSourceTask,
        ProvenanceHeaderEnabled: &provenanceHeaderEnabledSourceTask,
        ConsumerStrategy: &consumerStrategySourceTask,
        CompressionType: &compressionTypeSourceTask,
        TopicsMapping: &topicsMappingSourceTask,
    }
    sourceTypeCreateSmartConnectTaskReq:=
```

```
model.GetCreateSmartConnectTaskReqSourceTypeEnum().KAFKA_REPLICATOR_SOURCE
startLaterCreateSmartConnectTaskReq:= true
taskNameCreateSmartConnectTaskReq:= "smart-connect-2"
request.Body = &model.CreateSmartConnectTaskReq{
    SourceTask: sourceTaskbody,
    SourceType: &sourceTypeCreateSmartConnectTaskReq,
    StartLater: &startLaterCreateSmartConnectTaskReq,
    TaskName: &taskNameCreateSmartConnectTaskReq,
}
response, err := client.CreateConnectorTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- Creating a Redis data replication task that starts immediately: full synchronization, 10 maximum retry times, 10,000 ms retry interval, and 10 KB/s bandwidth limit.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    redisAddressSinkTask:= "192.168.119.51:6379"
    redisTypeSinkTask:= "standalone"
    redisPasswordSinkTask:= "*****"
    sinkTaskbody := &model.SmartConnectTaskReqSinkConfig{
        RedisAddress: &redisAddressSinkTask,
        RedisType: &redisTypeSinkTask,
        RedisPassword: &redisPasswordSinkTask,
    }
    redisAddressSourceTask:= "192.168.99.249:6379,192.168.120.127:6379,192.168.116.219:6379"
    redisTypeSourceTask:= "cluster"
    redisPasswordSourceTask:= "*****"
    syncModeSourceTask:= "RDB_ONLY"
    fullSyncWaitMsSourceTask:= int32(10000)
    fullSyncMaxRetrySourceTask:= int32(10)
```

```

ratelimitSourceTask:= int32(10)
sourceTaskbody := &model.SmartConnectTaskReqSourceConfig{
    RedisAddress: &redisAddressSourceTask,
    RedisType: &redisTypeSourceTask,
    RedisPassword: &redisPasswordSourceTask,
    SyncMode: &syncModeSourceTask,
    FullSyncWaitMs: &fullSyncWaitMsSourceTask,
    FullSyncMaxRetry: &fullSyncMaxRetrySourceTask,
    Ratelimit: &ratelimitSourceTask,
}
sinkTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSinkTypeEnum().REDIS_REPLICATOR_SINK
sourceTypeCreateSmartConnectTaskReq:=
model.GetCreateSmartConnectTaskReqSourceTypeEnum().REDIS_REPLICATOR_SOURCE
topicsCreateSmartConnectTaskReq:= "topic-sc-3"
startLaterCreateSmartConnectTaskReq:= false
taskNameCreateSmartConnectTaskReq:= "smart-connect-3"
request.Body = &model.CreateSmartConnectTaskReq{
    SinkTask: sinkTaskbody,
    SinkType: &sinkTypeCreateSmartConnectTaskReq,
    SourceTask: sourceTaskbody,
    SourceType: &sourceTypeCreateSmartConnectTaskReq,
    Topics: &topicsCreateSmartConnectTaskReq,
    StartLater: &startLaterCreateSmartConnectTaskReq,
    TaskName: &taskNameCreateSmartConnectTaskReq,
}
response, err := client.CreateConnectorTask(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful.

Error Codes

See [Error Codes](#).

5.3.4 Querying Smart Connect Tasks

Function

This API is used to query Smart Connect tasks.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/connector/tasks

Table 5-96 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 5-97 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	Integer	Maximum number of instances returned in the current query. The default value is 10 . The value ranges from 1 to 50.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-98 Response body parameters

Parameter	Type	Description
tasks	Array of SmartConnectTaskEntity objects	Smart Connect task details.
total_number	Integer	Number of Smart Connect tasks.
max_tasks	Integer	Maximum number of Smart Connect tasks.
quota_tasks	Integer	Smart Connect task quota.

Table 5-99 SmartConnectTaskEntity

Parameter	Type	Description
task_name	String	Smart Connect task name.
topics	String	Topic of a Smart Connect task.
topics_regex	String	Regular expression of the topic of a Smart Connect task.
source_type	String	Source type of a Smart Connect task.
source_task	SmartConnectTaskRespSourceConfig object	Source configuration of a Smart Connect task.
sink_type	String	Target type of a Smart Connect task.
sink_task	SmartConnectTaskRespSinkConfig object	Target type of a Smart Connect task.
id	String	ID of a Smart Connect task.
status	String	Smart Connect task status.
create_time	Long	Time when the Smart Connect task was created.

Table 5-100 SmartConnectTaskRespSourceConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the source type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the source type is Redis.)
dcs_instance_id	String	DCS instance ID. (Displayed only when the source type is Redis.)
sync_mode	String	Synchronization type: RDB_ONLY indicates full synchronization; CUSTOM_OFFSET indicates full and incremental synchronization. (Displayed only when the source type is Redis.)
full_sync_wait_ms	Integer	Interval of full synchronization retries, in ms. (Displayed only when the source type is Redis.)

Parameter	Type	Description
full_sync_max_retry	Integer	Max. retries of full synchronization. (Displayed only when the source type is Redis.)
ratelimit	Integer	Rate limit, in KB/s. -1: disable. (Displayed only when the source type is Redis.)
current_cluster_name	String	Current Kafka instance name. (Displayed only when the source type is Kafka.)
cluster_name	String	Target Kafka instance name. (Displayed only when the source type is Kafka.)
user_name	String	Username of the target Kafka instance. (Displayed only when the source type is Kafka.)
sasl_mechanism	String	Target Kafka authentication mode. (Displayed only when the source type is Kafka.)
instance_id	String	Target Kafka instance ID. (Displayed only when the source type is Kafka.)
bootstrap_servers	String	Target Kafka instance address. (Displayed only when the source type is Kafka.)
security_protocol	String	Target Kafka authentication. (Displayed only when the source type is Kafka.)
direction	String	Sync direction. (Displayed only when the source type is Kafka.)
sync_consumer_of_fsets_enabled	Boolean	Indicates whether to sync the consumption progress. (Displayed only when the source type is Kafka.)
replication_factor	Integer	Number of replicas. (Displayed only when the source type is Kafka.)
task_num	Integer	Number of tasks. (Displayed only when the source type is Kafka.)
rename_topic_enabled	Boolean	Indicates whether to rename a topic. (Displayed only when the source type is Kafka.)
provenance_header_enabled	Boolean	Indicates whether to add the source header. (Displayed only when the source type is Kafka.)

Parameter	Type	Description
consumer_strategy	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the source type is Kafka.)
compression_type	String	Compression algorithm. (Displayed only when the source type is Kafka.)
topics_mapping	String	Topic mapping. (Displayed only when the source type is Kafka.)

Table 5-101 SmartConnectTaskRespSinkConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the target type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the target type is Redis.)
dcs_instance_id	String	DCS instance ID. (Displayed only when the target type is Redis.)
target_db	Integer	Target database. The default value is -1 . (Displayed only when the target type is Redis.)
consumer_strategy	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the target type is OBS.)
destination_file_type	String	Dump file format. Only TEXT is supported. (Displayed only when the target type is OBS.)
deliver_time_interval	Integer	Dumping period (s). (Displayed only when the target type is OBS.)
obs_bucket_name	String	Dumping address. (Displayed only when the target type is OBS.)
obs_path	String	Dump directory. (Displayed only when the target type is OBS.)
partition_format	String	Time directory format. (Displayed only when the target type is OBS.)
record_delimiter	String	Line break. (Displayed only when the target type is OBS.)

Parameter	Type	Description
store_keys	Boolean	Storage key. (Displayed only when the target type is OBS.)
obs_part_size	Integer	Size (in bytes) of each file to be uploaded. The default value is 5242880 . (Displayed only when the target type is OBS.)
flush_size	Integer	flush_size. (Displayed only when the target type is OBS.)
timezone	String	Time zone. (Displayed only when the target type is OBS.)
schema_generator_class	String	schema_generator class. The default value is io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Displayed only when the target type is OBS.)
partitioner_class	String	partitioner class. The default value is io.confluent.connect.storage.partitionner.TimeBasedPartitioner . (Displayed only when the target type is OBS.)
value_converter	String	value_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)
key_converter	String	key_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)
kv_delimiter	String	kv_delimiter. The default value is . . (Displayed only when the target type is OBS.)

Example Requests

None

Example Responses

Status code: 200

Successful.

```
{
  "tasks": [ {
    "task_name": "smart-connect-1571576841",
```

```

"topics" : "topic-1643449744",
"source_task" : {
  "current_cluster_name" : "A",
  "cluster_name" : "B",
  "direction" : "pull",
  "bootstrap_servers" : "192.168.45.58:9092,192.168.44.1:9092,192.168.41.230:9092,192.168.43.112:9092",
  "instance_id" : "59f6d088-****-****-****-*****",
  "consumer_strategy" : "earliest",
  "sync_consumer_offsets_enabled" : false,
  "rename_topic_enabled" : false,
  "provenance_header_enabled" : false,
  "security_protocol" : "PLAINTEXT",
  "saslm_mechanism" : "PLAIN",
  "user_name" : "",
  "topics_mapping" : "",
  "compression_type" : "none",
  "task_num" : 2,
  "replication_factor" : 3
},
"source_type" : "KAFKA_REPLICATOR_SOURCE",
"sink_task" : null,
"sink_type" : "NONE",
"id" : "194917d0-****-****-****-*****",
"status" : "RUNNING",
"create_time" : 1708427753133
}, {
  "task_name" : "smart-connect-1",
  "topics_regex" : "topic-obs*",
  "source_task" : null,
  "source_type" : "NONE",
  "sink_task" : {
    "consumer_strategy" : "earliest",
    "destination_file_type" : "TEXT",
    "obs_bucket_name" : "abcabc",
    "obs_path" : "obsTransfer-1810125534",
    "partition_format" : "yyyy/MM/dd/HH/mm",
    "record_delimiter" : "\\n",
    "deliver_time_interva" : 300,
    "obs_part_size" : 5242880,
    "flush_size" : 1000000,
    "timezone" : "Asia/Chongqing",
    "schema_generator_class" : "io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator",
    "partitioner_class" : "io.confluent.connect.storage.partitioner.TimeBasedPartitioner",
    "value_converter" : "org.apache.kafka.connect.converters.ByteArrayConverter",
    "key_converter" : "org.apache.kafka.connect.converters.ByteArrayConverter",
    "store_keys" : false,
    "kv_delimiter" : ":"
  },
  "sink_type" : "OBS_SINK",
  "id" : "3c0ac4d1-****-****-****-*****",
  "status" : "RUNNING",
  "create_time" : 1708565483911
}, {
  "task_name" : "smart-connect-121248117",
  "topics" : "topic-sc",
  "source_task" : {
    "redis_address" : "192.168.91.179:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3cae1e7cd",
    "sync_mode" : "RDB_ONLY",
    "full_sync_wait_ms" : 13000,
    "full_sync_max_retry" : 4,
    "ratelimit" : -1
  },
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",

```

```
"target_db" : -1
},
"sink_type" : "REDIS_REPLICATOR_SINK",
"id" : "8a205bbd-****-****-****-*****",
"status" : "RUNNING",
"create_time" : 1708427753133
}],
"total_number" : 3,
"max_tasks" : 18,
"quota_tasks" : 18
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListConnectorTasksSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListConnectorTasksRequest request = new ListConnectorTasksRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListConnectorTasksResponse response = client.listConnectorTasks(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkkafka.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = KafkaClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListConnectorTasksRequest()  
        request.instance_id = "{instance_id}"  
        response = client.list_connector_tasks(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()
```



```

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListConnectorTasksRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListConnectorTasks(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful.

Error Codes

See [Error Codes](#).

5.3.5 Querying Smart Connect Task Details

Function

This API is used to query Smart Connect task details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}

Table 5-102 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Parameter	Mandatory	Type	Description
task_id	Yes	String	Smart Connect task ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-103 Response body parameters

Parameter	Type	Description
task_name	String	Smart Connect task name.
topics	String	Topic of a Smart Connect task.
topics_regex	String	Regular expression of the topic of a Smart Connect task.
source_type	String	Source type of a Smart Connect task.
source_task	SmartConnectTaskRespSourceConfig object	Source configuration of a Smart Connect task.
sink_type	String	Target type of a Smart Connect task.
sink_task	SmartConnectTaskRespSinkConfig object	Target type of a Smart Connect task.
id	String	ID of a Smart Connect task.
status	String	Smart Connect task status.
create_time	Long	Time when the Smart Connect task was created.

Table 5-104 SmartConnectTaskRespSourceConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the source type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the source type is Redis.)

Parameter	Type	Description
dcs_instance_id	String	DCS instance ID. (Displayed only when the source type is Redis.)
sync_mode	String	Synchronization type: RDB_ONLY indicates full synchronization; CUSTOM_OFFSET indicates full and incremental synchronization. (Displayed only when the source type is Redis.)
full_sync_wait_ms	Integer	Interval of full synchronization retries, in ms. (Displayed only when the source type is Redis.)
full_sync_max_retry	Integer	Max. retries of full synchronization. (Displayed only when the source type is Redis.)
ratelimit	Integer	Rate limit, in KB/s. -1 : disable. (Displayed only when the source type is Redis.)
current_cluster_name	String	Current Kafka instance name. (Displayed only when the source type is Kafka.)
cluster_name	String	Target Kafka instance name. (Displayed only when the source type is Kafka.)
user_name	String	Username of the target Kafka instance. (Displayed only when the source type is Kafka.)
sasl_mechanism	String	Target Kafka authentication mode. (Displayed only when the source type is Kafka.)
instance_id	String	Target Kafka instance ID. (Displayed only when the source type is Kafka.)
bootstrap_servers	String	Target Kafka instance address. (Displayed only when the source type is Kafka.)
security_protocol	String	Target Kafka authentication. (Displayed only when the source type is Kafka.)
direction	String	Sync direction. (Displayed only when the source type is Kafka.)

Parameter	Type	Description
sync_consumer_of fsets_enabled	Boolean	Indicates whether to sync the consumption progress. (Displayed only when the source type is Kafka.)
replication_factor	Integer	Number of replicas. (Displayed only when the source type is Kafka.)
task_num	Integer	Number of tasks. (Displayed only when the source type is Kafka.)
rename_topic_ena bled	Boolean	Indicates whether to rename a topic. (Displayed only when the source type is Kafka.)
provenance_head er_enabled	Boolean	Indicates whether to add the source header. (Displayed only when the source type is Kafka.)
consumer_strateg y	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the source type is Kafka.)
compression_type	String	Compression algorithm. (Displayed only when the source type is Kafka.)
topics_mapping	String	Topic mapping. (Displayed only when the source type is Kafka.)

Table 5-105 SmartConnectTaskRespSinkConfig

Parameter	Type	Description
redis_address	String	Redis instance address. (Displayed only when the target type is Redis.)
redis_type	String	Redis instance type. (Displayed only when the target type is Redis.)
dcs_instance_id	String	DCS instance ID. (Displayed only when the target type is Redis.)
target_db	Integer	Target database. The default value is -1 . (Displayed only when the target type is Redis.)
consumer_strateg y	String	Start offset. latest : Obtain the latest data; earliest : Obtain the earliest data. (Displayed only when the target type is OBS.)

Parameter	Type	Description
destination_file_type	String	Dump file format. Only TEXT is supported. (Displayed only when the target type is OBS.)
deliver_time_interval	Integer	Dumping period (s). (Displayed only when the target type is OBS.)
obs_bucket_name	String	Dumping address. (Displayed only when the target type is OBS.)
obs_path	String	Dump directory. (Displayed only when the target type is OBS.)
partition_format	String	Time directory format. (Displayed only when the target type is OBS.)
record_delimiter	String	Line break. (Displayed only when the target type is OBS.)
store_keys	Boolean	Storage key. (Displayed only when the target type is OBS.)
obs_part_size	Integer	Size (in bytes) of each file to be uploaded. The default value is 5242880 . (Displayed only when the target type is OBS.)
flush_size	Integer	flush_size. (Displayed only when the target type is OBS.)
timezone	String	Time zone. (Displayed only when the target type is OBS.)
schema_generator_class	String	schema_generator class. The default value is io.confluent.connect.storage.hive.schema.DefaultSchemaGenerator . (Displayed only when the target type is OBS.)
partitioner_class	String	partitioner class. The default value is io.confluent.connect.storage.partitioners.TimeBasedPartitioner . (Displayed only when the target type is OBS.)
value_converter	String	value_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)
key_converter	String	key_converter. The default value is org.apache.kafka.connect.converters.ByteArrayConverter . (Displayed only when the target type is OBS.)

Parameter	Type	Description
kv_delimiter	String	kv_delimiter. The default value is :. (Displayed only when the target type is OBS.)

Example Requests

None

Example Responses

Status code: 200

Successful.

```
{
  "task_name" : "smart-connect-121248117",
  "topics" : "topic-sc",
  "source_task" : {
    "redis_address" : "192.168.91.179:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "949190a2-598a-4afd-99a8-dad3cae1e7cd",
    "sync_mode" : "RDB_ONLY",
    "full_sync_wait_ms" : 13000,
    "full_sync_max_retry" : 4,
    "ratelimit" : -1
  },
  "source_type" : "REDIS_REPLICATOR_SOURCE",
  "sink_task" : {
    "redis_address" : "192.168.119.51:6379",
    "redis_type" : "standalone",
    "dcs_instance_id" : "9b981368-a8e3-416a-87d9-1581a968b41b",
    "target_db" : -1
  },
  "sink_type" : "REDIS_REPLICATOR_SINK",
  "id" : "8a205bbd-7181-4b5e-9bd6-37274ce84577",
  "status" : "RUNNING",
  "create_time" : 1708427753133
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowConnectorTaskSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    ShowConnectorTaskRequest request = new ShowConnectorTaskRequest();
    request.withInstanceId("{instance_id}");
    request.withTaskId("{task_id}");
    try {
        ShowConnectorTaskResponse response = client.showConnectorTask(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowConnectorTaskRequest()
        request.instance_id = "{instance_id}"
```

```
request.task_id = "{task_id}"
response = client.show_connector_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful.

Error Codes

See [Error Codes](#).

5.3.6 Deleting a Smart Connect Task

Function

This API is used to delete a Smart Connect task.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}

Table 5-106 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
task_id	Yes	String	Smart Connect task ID.

Request Parameters

None

Response Parameters

None

Example Requests

```
DELETE https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteConnectorTaskRequest request = new DeleteConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            DeleteConnectorTaskResponse response = client.deleteConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteConnectorTaskRequest()
    request.instance_id = "{instance_id}"
    request.task_id = "{task_id}"
    response = client.delete_connector_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.DeleteConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Successful.

Error Codes

See [Error Codes](#).

5.3.7 Pausing a Smart Connect Task

Function

This API is used to pause a Smart Connect task.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/pause

Table 5-107 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
task_id	Yes	String	Smart Connect task ID.

Request Parameters

None

Response Parameters

None

Example Requests

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/pause
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class PauseConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        PauseConnectorTaskRequest request = new PauseConnectorTaskRequest();
        request.withInstanceId("{instance_id}");
        request.withTaskId("{task_id}");
        try {
            PauseConnectorTaskResponse response = client.pauseConnectorTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = PauseConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.pause_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.PauseConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.PauseConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Successful.

Error Codes

See [Error Codes](#).

5.3.8 Restarting a Smart Connect Task

Function

This API is used to restart a Smart Connect task.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/resume

Table 5-108 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Parameter	Mandatory	Type	Description
task_id	Yes	String	Smart Connect task ID.

Request Parameters

None

Response Parameters

None

Example Requests

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/resume
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResumeConnectorTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResumeConnectorTaskRequest request = new ResumeConnectorTaskRequest();
```



```
request.withInstanceId("{instance_id}");
request.withTaskId("{task_id}");
try {
    ResumeConnectorTaskResponse response = client.resumeConnectorTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResumeConnectorTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.resume_connector_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResumeConnectorTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ResumeConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Successful.

Error Codes

See [Error Codes](#).

5.3.9 Starting a Smart Connect Task or Restarting a Paused or Running Smart Connect Task

Function

This API is used to **start a Smart Connect task** or **restart a paused or running Smart Connect task**. Note that the sync progress will reset and the task will restart.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/kafka/instances/{instance_id}/connector/tasks/{task_id}/restart

Table 5-109 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
task_id	Yes	String	Smart Connect task ID.

Request Parameters

None

Response Parameters

None

Example Requests

```
PUT https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/connector/tasks/{task_id}/restart
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class RestartConnectorTaskSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    RestartConnectorTaskRequest request = new RestartConnectorTaskRequest();
    request.withInstanceId("{instance_id}");
    request.withTaskId("{task_id}");
    try {
        RestartConnectorTaskResponse response = client.restartConnectorTask(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RestartConnectorTaskRequest()
        request.instance_id = "{instance_id}"
```

```
request.task_id = "{task_id}"
response = client.restart_connector_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.RestartConnectorTaskRequest{
        request.InstanceId = "{instance_id}"
        request.TaskId = "{task_id}"
    }
    response, err := client.RestartConnectorTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Successful.

Error Codes

See [Error Codes](#).

5.4 Specification Modification Management

5.4.1 Increasing Instance Specifications

Function

This API is used to modify instance specifications. Currently, this API can only be used to increase specifications of pay-per-use instances.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{engine}/{project_id}/instances/{instance_id}/extend

Table 5-110 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-111 Request body parameters

Parameter	Mandatory	Type	Description
oper_type	Yes	String	Change type. Value range: <ul style="list-style-type: none"> ● storage: Expand the storage without changing the broker quantity. ● horizontal: Add brokers without resizing the storage space of each broker. ● vertical: Scale up the broker flavor without changing the broker quantity and storage.
new_storage_space	No	Integer	New storage space. This parameter is valid and mandatory when oper_type is set to storage or horizontal . Instance storage space = Number of brokers × Storage space of each broker. If oper_type is set to storage , the number of brokers remains unchanged, and the storage space of each broker must be expanded by at least 100 GB. If oper_type is set to horizontal , the storage space of each broker remains unchanged.
new_broker_number	No	Integer	This parameter is valid only when oper_type is set to horizontal . Value range: A maximum of 30 brokers are supported.
new_product_id	No	String	New product ID for scale-up. This parameter is valid and mandatory when oper_type is set to vertical . Obtain the product ID from Querying Product Specifications List .

Parameter	Mandatory	Type	Description
publicip_id	No	String	ID of the EIP bound to the instance. Use commas (,) to separate multiple EIP IDs. This parameter is mandatory when oper_type is set to horizontal .
tenant_ips	No	Array of strings	Specified IPv4 private IP addresses. The number of specified IP addresses must be less than or equal to the number of new brokers. If the number of specified IP addresses is less than the number of brokers, the unspecified brokers are randomly assigned private IP addresses.
second_tenant_subnet_id	No	String	New brokers use the standby subnet ID in instance expansion. This value is transferred when a standby subnet is used in instance expansion. Contact customer service to use the value.

Response Parameters

Status code: 200

Table 5-112 Response body parameters

Parameter	Type	Description
job_id	String	ID of the specification modification task.

Example Requests

- Expanding the storage space (pay-per-use)
POST `https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend`
{


```
"oper_type" : "storage",  
"new_storage_space" : 600  
}
```

- Adding brokers (pay-per-use)

POST https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend

```
{  
  "oper_type" : "horizontal",  
  "new_storage_space" : 1600,  
  "new_broker_num" : 4,  
  "tenant_ips" : [ "127.xx.xx.x", "127.xx.xx.x", "127.xx.xx.x" ]  
}
```

- Increasing the broker flavor (pay-per-use)

POST https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend

```
{  
  "oper_type" : "vertical",  
  "new_product_id" : "c6.4u8g.cluster"  
}
```

Example Responses

Status code: 200

Instance specifications increased.

```
{  
  "job_id" : "93b94287-728d-4bb1-a158-cb66cb0854e7"  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

- Expanding the storage space (pay-per-use)

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ResizeEngineInstanceSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before  
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local  
        // environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)
```

```
        .withAk(ak)
        .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
request.withInstanceId("{instance_id}");
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
body.withNewStorageSpace(600);
body.withOperType("storage");
request.withBody(body);
try {
    ResizeEngineInstanceResponse response = client.resizeEngineInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- Adding brokers (pay-per-use)

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ResizeEngineInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
    }
}
```

```
request.withInstanceId("{instance_id}");
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
List<String> listbodyTenantIps = new ArrayList<>();
listbodyTenantIps.add("127.xx.xx.x");
listbodyTenantIps.add("127.xx.xx.x");
listbodyTenantIps.add("127.xx.xx.x");
body.withTenantIps(listbodyTenantIps);
body.withNewBrokerNum(4);
body.withNewStorageSpace(1600);
body.withOperType("horizontal");
request.withBody(body);
try {
    ResizeEngineInstanceResponse response = client.resizeEngineInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **Increasing the broker flavor (pay-per-use)**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResizeEngineInstanceSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
        request.withInstanceId("{instance_id}");
        ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
        body.withNewProductId("c6.4u8g.cluster");
        body.withOperType("vertical");
        request.withBody(body);
```

```
try {
    ResizeEngineInstanceResponse response = client.resizeEngineInstance(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- Expanding the storage space (pay-per-use)

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeEngineInstanceRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        request.body = ResizeEngineInstanceReq(
            new_storage_space=600,
            oper_type="storage"
        )
        response = client.resize_engine_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Adding brokers (pay-per-use)

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResizeEngineInstanceRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        listTenantIpsbody = [
            "127.xx.xx.x",
            "127.xx.xx.x",
            "127.xx.xx.x"
        ]
        request.body = ResizeEngineInstanceReq(
            tenant_ips=listTenantIpsbody,
            new_broker_num=4,
            new_storage_space=1600,
            oper_type="horizontal"
        )
        response = client.resize_engine_instance(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Increasing the broker flavor (pay-per-use)

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ResizeEngineInstanceRequest()
    request.engine = "{engine}"
    request.instance_id = "{instance_id}"
    request.body = ResizeEngineInstanceReq(
        new_product_id="c6.4u8g.cluster",
        oper_type="vertical"
    )
    response = client.resize_engine_instance(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Expanding the storage space (pay-per-use)

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{
        request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
        request.InstanceId = "{instance_id}"
        newStorageSpaceResizeEngineInstanceReq:= int32(600)
        request.Body = &model.ResizeEngineInstanceReq{
            NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
            OperType: "storage",
        }
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Adding brokers (pay-per-use)

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{}
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    request.InstanceId = "{instance_id}"
    var listTenantIpsbody = []string{
        "127.xx.xx.x",
        "127.xx.xx.x",
        "127.xx.xx.x",
    }
    newBrokerNumResizeEngineInstanceReq:= int32(4)
    newStorageSpaceResizeEngineInstanceReq:= int32(1600)
    request.Body = &model.ResizeEngineInstanceReq{
        TenantIps: &listTenantIpsbody,
        NewBrokerNum: &newBrokerNumResizeEngineInstanceReq,
        NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
        OperType: "horizontal",
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- Increasing the broker flavor (pay-per-use)

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

```

```

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{}
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    request.InstanceId = "{instance_id}"
    newProductIdResizeEngineInstanceReq:= "c6.4u8g.cluster"
    request.Body = &model.ResizeEngineInstanceReq{
        NewProductId: &newProductIdResizeEngineInstanceReq,
        OperType: "vertical",
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Instance specifications increased.

Error Codes

See [Error Codes](#).

5.4.2 Querying Product Information for Instance Specification Modification

Function

This API is used to query the product information for instance specification modification.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/extend

Table 5-113 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 5-114 Query Parameters

Parameter	Mandatory	Type	Description
type	Yes	String	Product edition. <ul style="list-style-type: none"> advanced: premium edition

Request Parameters

None

Response Parameters

Status code: 200

Table 5-115 Response body parameters

Parameter	Type	Description
engine	String	Message engine: Kafka.

Parameter	Type	Description
versions	Array of strings	Versions supported by the message engine.
products	Array of ExtendProductInfoEntity objects	Product information for specification modification.

Table 5-116 ExtendProductInfoEntity

Parameter	Type	Description
type	String	Instance type.
product_id	String	Product ID.
ecs_flavor_id	String	ECS flavor used by the product.
arch_types	Array of strings	Supported CPU architectures.
charging_mode	Array of strings	Supported billing modes.
ios	Array of ExtendProductIosEntity objects	Disk I/O information.
support_features	Array of ExtendProductSupportFeaturesEntity objects	Supported features.
properties	ExtendProductPropertiesEntity object	Product specification description.
available_zones	Array of strings	AZs where there are available resources.
unavailable_zones	Array of strings	AZs where resources are sold out.

Table 5-117 ExtendProductIosEntity

Parameter	Type	Description
io_spec	String	Storage I/O specification.
available_zones	Array of strings	AZs where there are available resources.
type	String	I/O type.
unavailable_zones	Array of strings	AZs where resources are sold out.

Table 5-118 ExtendProductSupportFeaturesEntity

Parameter	Type	Description
name	String	Feature name.
properties	Map<String,String >	Key-value pair of a feature.

Table 5-119 ExtendProductPropertiesEntity

Parameter	Type	Description
max_partition_per_broker	String	Maximum number of partitions of each broker.
max_broker	String	Maximum number of brokers.
max_storage_per_node	String	Maximum storage space of each broker. Unit: GB.
max_consumer_per_broker	String	Maximum number of consumers of each broker.
min_broker	String	Minimum number of brokers.
max_bandwidth_per_broker	String	Maximum bandwidth of each broker.
min_storage_per_node	String	Minimum storage space of each broker. Unit: GB.
max_tps_per_broker	String	Maximum TPS of each broker.
product_alias	String	Alias of product_id .

Example Requests

Querying product information for instance specification modification

```
GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/extend?type={type}
```

Example Responses

Status code: 200

Successfully queried the product information for instance specification modification.

```
{
  "engine": "kafka",
  "versions": [ "1.1.0", "2.7" ],
  "products": [ {
    "type": "cluster",
    "product_id": "c6.2u4g.cluster",
```

```
"ecs_flavor_id" : "c3.large.2",
"arch_types" : [ "X86" ],
"charging_mode" : [ "monthly", "hourly" ],
"ios" : [ {
  "io_spec" : "dms.physical.storage.high.v2",
  "available_zones" : [ "xxx" ],
  "type" : "evs",
  "unavailable_zones" : [ ]
}, {
  "io_spec" : "dms.physical.storage.ultra.v2",
  "available_zones" : [ "xxx" ],
  "type" : "evs",
  "unavailable_zones" : [ ]
}],
"support_features" : [ {
  "name" : "connector_obs",
  "properties" : {
    "max_task" : "10",
    "max_node" : "10",
    "min_task" : "1",
    "min_node" : "2"
  }
}],
"properties" : {
  "max_partition_per_broker" : "250",
  "max_broker" : "30",
  "max_storage_per_node" : "10000",
  "max_consumer_per_broker" : "4000",
  "min_broker" : "3",
  "product_alias" : "kafka.2u4g.cluster",
  "max_bandwidth_per_broker" : "100",
  "min_storage_per_node" : "100",
  "max_tps_per_broker" : "30000"
},
"available_zones" : [ "xxx" ],
"unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.2u4g.cluster.dec",
  "ecs_flavor_id" : "c6.large.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.dss.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.dss.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }
],
"support_features" : [ {
  "name" : "connector_obs",
  "properties" : {
    "max_task" : "10",
    "max_node" : "10",
    "min_task" : "1",
```

```

    "min_node" : "2"
  }
}],
"properties" : {
  "max_partition_per_broker" : "100",
  "max_broker" : "30",
  "max_storage_per_node" : "400",
  "max_consumer_per_broker" : "4000",
  "min_broker" : "3",
  "product_alias" : "kafka.2u4g.cluster.dec",
  "max_bandwidth_per_broker" : "100",
  "min_storage_per_node" : "100",
  "max_tps_per_broker" : "30000"
},
"available_zones" : [ ],
"unavailable_zones" : [ "xxx" ]
}, {
  "type" : "cluster",
  "product_id" : "c6.4u8g.cluster",
  "ecs_flavor_id" : "c3.xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }
],
"support_features" : [ {
  "name" : "connector_obs",
  "properties" : {
    "max_task" : "10",
    "max_node" : "10",
    "min_task" : "1",
    "min_node" : "2"
  }
}
}],
"properties" : {
  "max_partition_per_broker" : "500",
  "max_broker" : "30",
  "max_storage_per_node" : "20000",
  "max_consumer_per_broker" : "4000",
  "min_broker" : "3",
  "product_alias" : "kafka.4u8g.cluster",
  "max_bandwidth_per_broker" : "100",
  "min_storage_per_node" : "100",
  "max_tps_per_broker" : "100000"
},
"available_zones" : [ "xxx" ],
"unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.8u16g.cluster",
  "ecs_flavor_id" : "c3.2xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],

```

```

    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "product_alias" : "kafka.8u16g.cluster",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "100",
    "max_tps_per_broker" : "150000"
  },
  "available_zones" : [ "xxx" ],
  "unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.12u24g.cluster",
  "ecs_flavor_id" : "c3.3xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1500",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "product_alias" : "kafka.12u24g.cluster",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "100",
    "max_tps_per_broker" : "200000"
  },
  "available_zones" : [ "xxx" ],
  "unavailable_zones" : [ ]
}, {
  "type" : "cluster",
  "product_id" : "c6.16u32g.cluster",
  "ecs_flavor_id" : "c3.4xlarge.2",
  "arch_types" : [ "X86" ],

```

```
"charging_mode" : [ "monthly", "hourly" ],
"ios" : [ {
  "io_spec" : "dms.physical.storage.high.v2",
  "available_zones" : [ "xxx" ],
  "type" : "evs",
  "unavailable_zones" : [ ]
}, {
  "io_spec" : "dms.physical.storage.ultra.v2",
  "available_zones" : [ "xxx" ],
  "type" : "evs",
  "unavailable_zones" : [ ]
} ],
"support_features" : [ {
  "name" : "connector_obs",
  "properties" : {
    "max_task" : "10",
    "max_node" : "10",
    "min_task" : "1",
    "min_node" : "2"
  }
} ],
"properties" : {
  "max_partition_per_broker" : "2000",
  "max_broker" : "30",
  "max_storage_per_node" : "30000",
  "max_consumer_per_broker" : "4000",
  "min_broker" : "3",
  "product_alias" : "kafka.16u32g.cluster",
  "max_bandwidth_per_broker" : "100",
  "min_storage_per_node" : "100",
  "max_tps_per_broker" : "250000"
},
"available_zones" : [ "xxx" ],
"unavailable_zones" : [ ]
} ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowEngineInstanceExtendProductInfoSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowEngineInstanceExtendProductInfoRequest request = new
ShowEngineInstanceExtendProductInfoRequest();

request.withEngine(ShowEngineInstanceExtendProductInfoRequest.EngineEnum.fromValue("{engine}"));
request.withInstanceId("{instance_id}");
try {
    ShowEngineInstanceExtendProductInfoResponse response =
client.showEngineInstanceExtendProductInfo(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEngineInstanceExtendProductInfoRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        response = client.show_engine_instance_extend_product_info(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```


Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEngineInstanceExtendProductInfoRequest{}
    request.Engine = model.GetShowEngineInstanceExtendProductInfoRequestEngineEnum().ENGINE
    request.InstanceId = "{instance_id}"
    response, err := client.ShowEngineInstanceExtendProductInfo(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successfully queried the product information for instance specification modification.

Error Codes

See [Error Codes](#).

5.5 Topic Management

5.5.1 Producing Messages to Kafka

Function

This API is used to send specified messages on the console to Kafka instances.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/messages/action

Table 5-120 Path Parameters

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

Table 5-121 Query Parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Action ID, which is send for production.

Request Parameters

Table 5-122 Request body parameters

Parameter	Mandatory	Type	Description
topic	Yes	String	Kafka topics.
body	Yes	String	Message content.
property_list	Yes	Array of property_list objects	Topic partition information.

Table 5-123 property_list

Parameter	Mandatory	Type	Description
name	No	String	Feature name.
value	No	String	Feature value.

Response Parameters

Status code: 200

Table 5-124 Response body parameters

Parameter	Type	Description
topic	String	Kafka topic.
body	String	Message content.
property_list	Array of objects	Topic partition information.

Example Requests

Sending messages on the Kafka console

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages/action?action_id={action_id}

```
{
  "topic": "XXXX",
  "body": "hello world",
  "property_list": [ {
    "name": "KEY",
    "value": "testKey"
  }, {
    "name": "PARTITION",
    "value": "0"
  } ]
}
```

Example Responses

Status code: 200

Message produced.

```
{
  "topic": "XXXX",
  "body": "XXXX",
  "property_list": [ ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Sending messages on the Kafka console

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class SendKafkaMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        SendKafkaMessageRequest request = new SendKafkaMessageRequest();
        request.withInstanceId("{instance_id}");
        SendKafkaMessageRequestBody body = new SendKafkaMessageRequestBody();
        List<SendKafkaMessageRequestBodyPropertyList> listbodyPropertyList = new ArrayList<>();
        listbodyPropertyList.add(
            new SendKafkaMessageRequestBodyPropertyList()
                .withName("KEY")
                .withValue("testKey")
        );
        listbodyPropertyList.add(
            new SendKafkaMessageRequestBodyPropertyList()
                .withName("PARTITION")
                .withValue("0")
        );
        body.withPropertyList(listbodyPropertyList);
        body.withBody("hello world");
        body.withTopic("XXXX");
        request.withBody(body);
        try {
            SendKafkaMessageResponse response = client.sendKafkaMessage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Sending messages on the Kafka console

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SendKafkaMessageRequest()
        request.instance_id = "{instance_id}"
        listPropertyListbody = [
            SendKafkaMessageRequestBodyPropertyList(
                name="KEY",
                value="testKey"
            ),
            SendKafkaMessageRequestBodyPropertyList(
                name="PARTITION",
                value="0"
            )
        ]
        request.body = SendKafkaMessageRequestBody(
            property_list=listPropertyListbody,
            body="hello world",
            topic="XXXX"
        )
        response = client.send_kafka_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Sending messages on the Kafka console

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SendKafkaMessageRequest{}
    request.InstanceId = "{instance_id}"
    namePropertyList := "KEY"
    valuePropertyList := "testKey"
    namePropertyList1 := "PARTITION"
    valuePropertyList1 := "0"
    var listPropertyListbody = []model.SendKafkaMessageRequestBodyPropertyList{
        {
            Name: &namePropertyList,
            Value: &valuePropertyList,
        },
        {
            Name: &namePropertyList1,
            Value: &valuePropertyList1,
        },
    }
    request.Body = &model.SendKafkaMessageRequestBody{
        PropertyList: listPropertyListbody,
        Body: "hello world",
        Topic: "XXXX",
    }
    response, err := client.SendKafkaMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Message produced.

Error Codes

See [Error Codes](#).

5.5.2 Creating a Topic for a Kafka Instance

Function

This API is used to create a topic for a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/topics

Table 5-125 Path Parameters

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

Request Parameters

Table 5-126 Request body parameters

Parameter	Mandatory	Type	Description
id	Yes	String	Topic name. A topic name consists of 3 to 200 characters, starts with a letter, and contains only letters, hyphens (-), underscores (_), periods (.), and digits.

Parameter	Mandatory	Type	Description
replication	No	Integer	Number of replicas, which is configured to ensure data reliability. Value range: 1 to 3.
sync_message_flush	No	Boolean	Whether synchronous flushing is enabled. The default value is false . Synchronous flushing compromises performance.
partition	No	Integer	Number of topic partitions, which is used to set the number of concurrently consumed messages. The value ranges from 1 to 200.
sync_replication	No	Boolean	Whether synchronous replication is enabled. After this function is enabled, the acks parameter on the producer client must be set to -1 . Otherwise, this parameter does not take effect.
retention_time	No	Integer	Retention period of a message. The default value is 72 . The value ranges from 1 to 720, in hours.
topic_other_configs	No	Array of topic_other_configs objects	Topic configuration.
topic_desc	No	String	Topic description.

Table 5-127 topic_other_configs

Parameter	Mandatory	Type	Description
name	No	String	Configuration name.
value	No	String	Configuration value.

Response Parameters

Status code: 200

Table 5-128 Response body parameters

Parameter	Type	Description
name	String	Topic name.

Example Requests

Creating a topic named test01

POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics

```
{
  "id" : "test01",
  "partition" : 3,
  "replication" : 3,
  "retention_time" : 72,
  "sync_message_flush" : false,
  "sync_replication" : "false",
  "topic_other_configs" : [ {
    "name" : "message.timestamp.type",
    "value" : "LogAppendTime"
  }, {
    "name" : "max.message.bytes",
    "value" : 10485760
  } ],
  "topic_desc" : ""
}
```

Example Responses

Status code: 200

Successful. A topic name is returned.

```
{
  "name" : "test01"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a topic named test01

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateInstanceTopicSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    KafkaClient client = KafkaClient.newBuilder()
        .withCredential(auth)
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
    CreateInstanceTopicRequest request = new CreateInstanceTopicRequest();
    request.withInstanceId("{instance_id}");
    CreateInstanceTopicReq body = new CreateInstanceTopicReq();
    List<CreateInstanceTopicReqTopicOtherConfigs> listbodyTopicOtherConfigs = new ArrayList<>();
    listbodyTopicOtherConfigs.add(
        new CreateInstanceTopicReqTopicOtherConfigs()
            .withName("message.timestamp.type")
            .withValue("LogAppendTime")
    );
    listbodyTopicOtherConfigs.add(
        new CreateInstanceTopicReqTopicOtherConfigs()
            .withName("max.message.bytes")
            .withValue("10485760")
    );
    body.withTopicDesc("");
    body.withTopicOtherConfigs(listbodyTopicOtherConfigs);
    body.withRetentionTime(72);
    body.withSyncReplication(false);
    body.withPartition(3);
    body.withSyncMessageFlush(false);
    body.withReplication(3);
    body.withId("test01");
    request.withBody(body);
    try {
        CreateInstanceTopicResponse response = client.createInstanceTopic(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

Creating a topic named test01

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudskkkafka.v2.region.kafka_region import KafkaRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateInstanceTopicRequest()
        request.instance_id = "{instance_id}"
        listTopicOtherConfigsbody = [
            CreateInstanceTopicReqTopicOtherConfigs(
                name="message.timestamp.type",
                value="LogAppendTime"
            ),
            CreateInstanceTopicReqTopicOtherConfigs(
                name="max.message.bytes",
                value="10485760"
            )
        ]
        request.body = CreateInstanceTopicReq(
            topic_desc="",
            topic_other_configs=listTopicOtherConfigsbody,
            retention_time=72,
            sync_replication=False,
            partition=3,
            sync_message_flush=False,
            replication=3,
            id="test01"
        )
        response = client.create_instance_topic(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Creating a topic named test01

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateInstanceTopicRequest{
    request.InstanceId = "{instance_id}"
    nameTopicOtherConfigs:= "message.timestamp.type"
    valueTopicOtherConfigs:= "LogAppendTime"
    nameTopicOtherConfigs1:= "max.message.bytes"
    valueTopicOtherConfigs1:= "10485760"
    var listTopicOtherConfigsboby = []model.CreateInstanceTopicReqTopicOtherConfigs{
        {
            Name: &nameTopicOtherConfigs,
            Value: &valueTopicOtherConfigs,
        },
        {
            Name: &nameTopicOtherConfigs1,
            Value: &valueTopicOtherConfigs1,
        },
    }
    topicDescCreateInstanceTopicReq:= ""
    retentionTimeCreateInstanceTopicReq:= int32(72)
    syncReplicationCreateInstanceTopicReq:= false
    partitionCreateInstanceTopicReq:= int32(3)
    syncMessageFlushCreateInstanceTopicReq:= false
    replicationCreateInstanceTopicReq:= int32(3)
    request.Body = &model.CreateInstanceTopicReq{
        TopicDesc: &topicDescCreateInstanceTopicReq,
        TopicOtherConfigs: &listTopicOtherConfigsboby,
        RetentionTime: &retentionTimeCreateInstanceTopicReq,
        SyncReplication: &syncReplicationCreateInstanceTopicReq,
        Partition: &partitionCreateInstanceTopicReq,
        SyncMessageFlush: &syncMessageFlushCreateInstanceTopicReq,
        Replication: &replicationCreateInstanceTopicReq,
        Id: "test01",
    }
}
response, err := client.CreateInstanceTopic(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful. A topic name is returned.

Error Codes

See [Error Codes](#).

5.5.3 Listing Topics of a Kafka Instance

Function

This API is used to query details about topics of a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/topics

Table 5-129 Path Parameters

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

Table 5-130 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	String	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	String	Maximum number of instances returned in the current query. The default value is 10 . The value ranges from 1 to 50.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-131 Response body parameters

Parameter	Type	Description
total	Integer	Total number of topics.
size	Integer	Maximum number of records to be displayed on a page.
remain_partitions	Integer	Number of remaining partitions.
max_partitions	Integer	Total number of partitions.
topic_max_partitions	Integer	Maximum number of partitions in a single topic.
topics	Array of TopicEntity objects	Topic list.

Table 5-132 TopicEntity

Parameter	Type	Description
policiesOnly	Boolean	Whether this policy is the default policy.
name	String	Topic name.
replication	Integer	Number of replicas, which is configured to ensure data reliability.
partition	Integer	Number of topic partitions, which is used to set the number of concurrently consumed messages.
retention_time	Integer	Retention period of a message.
sync_replication	Boolean	Whether synchronous replication is enabled. After this function is enabled, the acks parameter on the producer client must be set to -1 . Otherwise, this parameter does not take effect.

Parameter	Type	Description
sync_message_flush	Boolean	Whether synchronous flushing is enabled. The default value is false . Synchronous flushing compromises performance.
external_configs	Object	Extended configuration.
topic_type	Integer	Topic type. Options: 0 : common topic; 1 : system (internal) topic.
topic_other_configs	Array of topic_other_configs objects	Other topic configurations.
topic_desc	String	Topic description.
created_at	Long	Topic creation time.

Table 5-133 topic_other_configs

Parameter	Type	Description
name	String	Configuration name.
valid_values	String	Valid value.
default_value	String	Default value.
config_type	String	Configuration type. The value can be dynamic or static .
value	String	Configuration value.
value_type	String	Configuration value type.

Example Requests

Querying the topic list

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics?offset=0&limit=10
```

Example Responses

Status code: 200

The query is successful.

```
{
  "total": 1,
  "size": 0,
  "topics": [ {
    "policiesOnly": false,
    "name": "Topic-test01",
    "replication": 3,
```

```
"partition" : 3,
"retention_time" : 72,
"sync_replication" : "false",
"sync_message_flush" : "false",
"topic_other_configs" : [ {
  "name" : "max.message.bytes",
  "valid_values" : "[0...10485760]",
  "default_value" : "10485760",
  "config_type" : "dynamic",
  "value" : "10485760",
  "value_type" : "int"
}, {
  "name" : "message.timestamp.type",
  "valid_values" : "[CreateTime, LogAppendTime]",
  "default_value" : "LogAppendTime",
  "config_type" : "dynamic",
  "value" : "LogAppendTime",
  "value_type" : "string"
} ],
"external_configs" : { },
"topic_type" : 0,
"topic_desc" : "This is a test topic",
"created_at" : 1688112779916
} ],
"remain_partitions" : 294,
"max_partitions" : 300,
"topic_max_partitions" : 200
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListInstanceTopicsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
    }
}
```



```
ListInstanceTopicsRequest request = new ListInstanceTopicsRequest();
request.withInstanceId("{instance_id}");
try {
    ListInstanceTopicsResponse response = client.listInstanceTopics(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceTopicsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_instance_topics(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstanceTopicsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListInstanceTopics(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.5.4 Modifying Topics of a Kafka Instance

Function

This API is used to modify topics of a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/topics

Table 5-134 Path Parameters

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

Request Parameters

Table 5-135 Request body parameters

Parameter	Mandatory	Type	Description
topics	No	Array of topics objects	Topics that were modified.

Table 5-136 topics

Parameter	Mandatory	Type	Description
id	Yes	String	Topic name, which cannot be modified.
retention_time	No	Integer	Aging time in hour.
sync_replication	No	Boolean	Whether synchronous replication is enabled.
sync_message_flush	No	Boolean	Whether synchronous flushing is enabled.
new_partition_numbers	No	Integer	Number of the partitions.
new_partition_brokers	No	Array of integers	Specifying brokers for new partitions.
topic_other_configs	No	Array of topic_other_configs objects	Topic configuration.
topic_desc	No	String	Topic description.

Table 5-137 topic_other_configs

Parameter	Mandatory	Type	Description
name	No	String	Configuration name.
value	No	String	Configuration value.

Response Parameters

None

Example Requests

Modifying parameters of topic-1284340884. Specifically, change the aging time to 72 hours, the number of partitions to 6, timestamp to LogAppendTime, max. batch size to 10485760, specify new partitions on broker-1 and broker-2, and disable synchronous replication and flushing.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics
```

```
{
  "topics": [ {
    "id": "test01",
    "retention_time": 72,
    "sync_replication": false,
    "sync_message_flush": false,
    "new_partition_numbers": 6,
    "new_partition_brokers": [ 1, 2 ],
    "topic_other_configs": [ {
      "name": "message.timestamp.type",
      "value": "LogAppendTime"
    }, {
      "name": "max.message.bytes",
      "value": 10485760
    } ],
    "topic_desc": "This is a test topic"
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying parameters of topic-1284340884. Specifically, change the aging time to 72 hours, the number of partitions to 6, timestamp to LogAppendTime, max. batch size to 10485760, specify new partitions on broker-1 and broker-2, and disable synchronous replication and flushing.

```
package com.huaweicloud.sdk.test;
import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateInstanceTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        UpdateInstanceTopicRequest request = new UpdateInstanceTopicRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceTopicReq body = new UpdateInstanceTopicReq();
        List<UpdateInstanceTopicReqTopicOtherConfigs> listTopicsTopicOtherConfigs = new ArrayList<>();
        listTopicsTopicOtherConfigs.add(
            new UpdateInstanceTopicReqTopicOtherConfigs()
                .withName("message.timestamp.type")
                .withValue("LogAppendTime")
        );
        listTopicsTopicOtherConfigs.add(
            new UpdateInstanceTopicReqTopicOtherConfigs()
                .withName("max.message.bytes")
                .withValue("10485760")
        );
        List<Integer> listTopicsNewPartitionBrokers = new ArrayList<>();
        listTopicsNewPartitionBrokers.add(1);
        listTopicsNewPartitionBrokers.add(2);
        List<UpdateInstanceTopicReqTopics> listbodyTopics = new ArrayList<>();
        listbodyTopics.add(
            new UpdateInstanceTopicReqTopics()
                .withId("test01")
                .withRetentionTime(72)
                .withSyncReplication(false)
                .withSyncMessageFlush(false)
                .withNewPartitionNumbers(6)
                .withNewPartitionBrokers(listTopicsNewPartitionBrokers)
                .withTopicOtherConfigs(listTopicsTopicOtherConfigs)
                .withTopicDesc("This is a test topic")
        );
        body.withTopics(listbodyTopics);
        request.withBody(body);
        try {
            UpdateInstanceTopicResponse response = client.updateInstanceTopic(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
```

```
e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Modifying parameters of topic-1284340884. Specifically, change the aging time to 72 hours, the number of partitions to 6, timestamp to LogAppendTime, max. batch size to 10485760, specify new partitions on broker-1 and broker-2, and disable synchronous replication and flushing.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceTopicRequest()
        request.instance_id = "{instance_id}"
        listTopicOtherConfigsTopics = [
            UpdateInstanceTopicReqTopicOtherConfigs(
                name="message.timestamp.type",
                value="LogAppendTime"
            ),
            UpdateInstanceTopicReqTopicOtherConfigs(
                name="max.message.bytes",
                value="10485760"
            )
        ]
        listNewPartitionBrokersTopics = [
            1,
            2
        ]
        listTopicsbody = [
            UpdateInstanceTopicReqTopics(
                id="test01",
                retention_time=72,
                sync_replication=False,
                sync_message_flush=False,
                new_partition_numbers=6,
                new_partition_brokers=listNewPartitionBrokersTopics,
```

```
        topic_other_configs=listTopicOtherConfigsTopics,  
        topic_desc="This is a test topic"  
    )  
]  
request.body = UpdateInstanceTopicReq(  
    topics=listTopicsbody  
)  
response = client.update_instance_topic(request)  
print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

Modifying parameters of topic-1284340884. Specifically, change the aging time to 72 hours, the number of partitions to 6, timestamp to LogAppendTime, max. batch size to 10485760, specify new partitions on broker-1 and broker-2, and disable synchronous replication and flushing.

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := kafka.NewKafkaClient(  
        kafka.KafkaClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateInstanceTopicRequest{}  
    request.InstanceId = "{instance_id}"  
    nameTopicOtherConfigs:= "message.timestamp.type"  
    valueTopicOtherConfigs:= "LogAppendTime"  
    nameTopicOtherConfigs1:= "max.message.bytes"  
    valueTopicOtherConfigs1:= "10485760"  
    var listTopicOtherConfigsTopics = []model.UpdateInstanceTopicReqTopicOtherConfigs{  
        {  
            Name: &nameTopicOtherConfigs,  
            Value: &valueTopicOtherConfigs,  
        },  
        {  
            Name: &nameTopicOtherConfigs1,  
            Value: &valueTopicOtherConfigs1,  
        },  
    }  
}
```

```
    },
  }
  var listNewPartitionBrokersTopics = []int32{
    int32(1),
    int32(2),
  }
  retentionTimeTopics:= int32(72)
  syncReplicationTopics:= false
  syncMessageFlushTopics:= false
  newPartitionNumbersTopics:= int32(6)
  topicDescTopics:= "This is a test topic"
  var listTopicsbody = []model.UpdateInstanceTopicReqTopics{
    {
      Id: "test01",
      RetentionTime: &retentionTimeTopics,
      SyncReplication: &syncReplicationTopics,
      SyncMessageFlush: &syncMessageFlushTopics,
      NewPartitionNumbers: &newPartitionNumbersTopics,
      NewPartitionBrokers: &listNewPartitionBrokersTopics,
      TopicOtherConfigs: &listTopicOtherConfigsTopics,
      TopicDesc: &topicDescTopics,
    },
  }
  request.Body = &model.UpdateInstanceTopicReq{
    Topics: &listTopicsbody,
  }
  response, err := client.UpdateInstanceTopic(request)
  if err == nil {
    fmt.Printf("%v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

More

For SDK sample code of more programming languages, see the [Sample Code](#) tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The modification is successful.

Error Codes

See [Error Codes](#).

5.5.5 Batch Deleting Topics of a Kafka Instance

Function

This API is used to delete multiple topics of a Kafka instance in batches. If some topics are deleted successfully while some fail to be deleted, a success response is returned with information about topics that fail to be deleted.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/topics/delete

Table 5-138 Path Parameters

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

Request Parameters

Table 5-139 Request body parameters

Parameter	Mandatory	Type	Description
topics	No	Array of strings	Topics to delete. This parameter is mandatory when instance topics are deleted in batches.

Response Parameters

Status code: 200

Table 5-140 Response body parameters

Parameter	Type	Description
topics	Array of topics objects	Topic list.

Table 5-141 topics

Parameter	Type	Description
id	String	Topic name.
success	Boolean	Whether the topics are deleted.

Example Requests

Batch deleting topics

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/topics/delete
{
  "topics" : [ "topic01" ]
}
```

Example Responses

Status code: 200

The deletion is successful.

```
{
  "topics" : [ {
    "id" : "topic01",
    "success" : true
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Batch deleting topics

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteInstanceTopicSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
```

```
BatchDeleteInstanceTopicRequest request = new BatchDeleteInstanceTopicRequest();
request.withInstanceId("{instance_id}");
BatchDeleteInstanceTopicReq body = new BatchDeleteInstanceTopicReq();
List<String> listbodyTopics = new ArrayList<>();
listbodyTopics.add("topic01");
body.withTopics(listbodyTopics);
request.withBody(body);
try {
    BatchDeleteInstanceTopicResponse response = client.batchDeleteInstanceTopic(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Batch deleting topics

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteInstanceTopicRequest()
        request.instance_id = "{instance_id}"
        listTopicsbody = [
            "topic01"
        ]
        request.body = BatchDeleteInstanceTopicReq(
            topics=listTopicsbody
        )
        response = client.batch_delete_instance_topic(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Batch deleting topics

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstanceTopicRequest{}
    request.InstanceId = "{instance_id}"
    var listTopicsbody = []string{
        "topic01",
    }
    request.Body = &model.BatchDeleteInstanceTopicReq{
        Topics: &listTopicsbody,
    }
    response, err := client.BatchDeleteInstanceTopic(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The deletion is successful.

Error Codes

See [Error Codes](#).

5.5.6 Querying the Partition List of a Topic

Function

This API is used to query the partition list of a topic.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/partitions

Table 5-142 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic.

Table 5-143 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset. The records after this offset will be queried.
limit	No	Integer	Maximum number of records that can be returned.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-144 Response body parameters

Parameter	Type	Description
total	Integer	Total records.
partitions	Array of partitions objects	Partition array.

Table 5-145 partitions

Parameter	Type	Description
partition	Integer	Partition ID.
start_offset	Long	Start offset.
last_offset	Long	Last offset.
message_count	Long	Number of messages in a partition.
last_update_time	Long	Last update time.

Example Requests

Querying the partition list of a topic

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/partitions?
start=1&limit=10
```

Example Responses

Status code: 200

The partition list of the topic is queried successfully.

```
{
  "total" : 3,
  "partitions" : [ {
    "partition" : 0,
    "start_offset" : 0,
    "last_offset" : 1216303,
    "message_count" : 1216303,
    "last_update_time" : 1688011291458
  }, {
    "partition" : 1,
    "start_offset" : 0,
    "last_offset" : 985447,
    "message_count" : 985447,
    "last_update_time" : 1688011291469
  }, {
    "partition" : 2,
    "start_offset" : 0,
    "last_offset" : 923340,
    "message_count" : 923340,
    "last_update_time" : 1688011291526
  }
  ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListTopicPartitionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTopicPartitionsRequest request = new ListTopicPartitionsRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        try {
            ListTopicPartitionsResponse response = client.listTopicPartitions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTopicPartitionsRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        response = client.list_topic_partitions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTopicPartitionsRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ListTopicPartitions(request)
    if err == nil {
```



```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The partition list of the topic is queried successfully.

Error Codes

See [Error Codes](#).

5.5.7 Querying the Current Producer List of a Topic

Function

This API is used to query the current producer list of a topic.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/producers

Table 5-146 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic.

Table 5-147 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset. The records after this offset will be queried.
limit	No	Integer	Maximum number of records that can be returned.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-148 Response body parameters

Parameter	Type	Description
total	Integer	Total records.
producers	Array of producers objects	Producer list.

Table 5-149 producers

Parameter	Type	Description
producer_address	String	Producer address.
broker_address	String	Broker address.
join_time	Long	Time when the broker was connected.

Example Requests

Querying the current producer list of a topic

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/producers?
offset=0&limit=10
```

Example Responses

Status code: 200

The current producer list of the topic is queried successfully.

```
{
  "total" : 3,
```

```
"producers" : [ {
  "producer_address" : "192.0.0.149:40443",
  "broker_address" : "192.0.0.146:9092",
  "join_time" : 1687204743328
}, {
  "producer_address" : "192.0.0.149:13807",
  "broker_address" : "192.0.0.80:9092",
  "join_time" : 1687204745939
}, {
  "producer_address" : "192.0.0.149:31876",
  "broker_address" : "192.0.0.71:9092",
  "join_time" : 1687204744934
} ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListTopicProducersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListTopicProducersRequest request = new ListTopicProducersRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        try {
            ListTopicProducersResponse response = client.listTopicProducers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```

```
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTopicProducersRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        response = client.list_topic_producers(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```

WithSk(sk).
WithProjectId(projectId).
Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListTopicProducersRequest{}
request.InstanceId = "{instance_id}"
request.Topic = "{topic}"
response, err := client.ListTopicProducers(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The current producer list of the topic is queried successfully.

Error Codes

See [Error Codes](#).

5.5.8 Querying Topic Details

Function

This API is used to query topic details of a Kafka instance. (Up to 1s for each instance call)

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}

Table 5-150 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-151 Response body parameters

Parameter	Type	Description
topic	String	Topic name.
partitions	Array of partitions objects	Partition list.
group_subscribed	Array of strings	List of consumer groups that subscribe to the topic.

Table 5-152 partitions

Parameter	Type	Description
partition	Integer	Partition ID.
leader	Integer	ID of the broker where the leader replica resides.
leo	Integer	LEO of the partition leader replica.
hw	Integer	High watermark (HW) of the partition.
lso	Integer	Log start offset (LSO) of the partition leader replica.
last_update_times tamp	Long	Time when the last message was written to the partition. The value is a Unix timestamp. Unit: ms

Parameter	Type	Description
replicas	Array of replicas objects	Replica list.

Table 5-153 replicas

Parameter	Type	Description
broker	Integer	ID of the broker where the replica resides.
leader	Boolean	Whether the replica is the leader.
in_sync	Boolean	Whether the replica is in the ISR.
size	Integer	Current log size of the replica. Unit: byte.
lag	Long	Number of messages that lag behind the high watermark in the replica.

Example Requests

Querying details about a specified topic

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}
```

Example Responses

Status code: 200

The query is successful.

```
{
  "topic": "test",
  "partitions": [ {
    "partition": 0,
    "leader": 2,
    "replicas": [ {
      "broker": 2,
      "leader": true,
      "in_sync": true,
      "size": 123971146,
      "lag": 0
    }, {
      "broker": 1,
      "leader": false,
      "in_sync": true,
      "size": 123971146,
      "lag": 0
    }, {
      "broker": 0,
      "leader": false,
      "in_sync": true,
      "size": 123971146,
      "lag": 0
    }
  ]
}
}
```

```
"lso" : 0,
"leo" : 13598,
"hw" : 13598,
"last_update_timestamp" : 1571477180985
}, {
  "partition" : 2,
  "leader" : 1,
  "replicas" : [ {
    "broker" : 1,
    "leader" : true,
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  }, {
    "broker" : 0,
    "leader" : false,
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  }, {
    "broker" : 2,
    "leader" : false,
    "in_sync" : true,
    "size" : 123889531,
    "lag" : 0
  } ],
  "lso" : 0,
  "leo" : 13601,
  "hw" : 13601,
  "last_update_timestamp" : 1571477077146
}, {
  "partition" : 1,
  "leader" : 0,
  "replicas" : [ {
    "broker" : 0,
    "leader" : true,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  }, {
    "broker" : 2,
    "leader" : false,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  }, {
    "broker" : 1,
    "leader" : false,
    "in_sync" : true,
    "size" : 127245604,
    "lag" : 0
  } ],
  "lso" : 0,
  "leo" : 13599,
  "hw" : 13599,
  "last_update_timestamp" : 1571477172959
} ],
"group_subscribed" : [ "test-consumer-group" ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceTopicDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        ShowInstanceTopicDetailRequest request = new ShowInstanceTopicDetailRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        try {
            ShowInstanceTopicDetailResponse response = client.showInstanceTopicDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowInstanceTopicDetailRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    response = client.show_instance_topic_detail(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceTopicDetailRequest{
        request.InstanceId = "{instance_id}"
        request.Topic = "{topic}"
    }
    response, err := client.ShowInstanceTopicDetail(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.6 Managing Consumer Groups

5.6.1 Querying Consumer Group Details

Function

This API is used to query consumer group details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/groups/{group}

Table 5-154 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-155 Response body parameters

Parameter	Type	Description
group	group object	Consumer group information.

Table 5-156 group

Parameter	Type	Description
group_id	String	Consumer group name.
state	String	Consumer group status. The value can be: <ul style="list-style-type: none"> • Dead: The consumer group has no members and no metadata. • Empty: The consumer group has metadata but has no members. • PreparingRebalance: The consumer group is to be rebalanced. • CompletingRebalance: All members have jointed the group. • Stable: Members in the consumer group can consume messages normally.
coordinator_id	Integer	Coordinator ID.
members	Array of members objects	Consumer list.
group_message_offsets	Array of group_message_offsets objects	Consumer offset.
assignment_strategy	String	Partition assignment policy.

Table 5-157 members

Parameter	Type	Description
host	String	Consumer address.

Parameter	Type	Description
assignment	Array of assignment objects	Details about the partition assigned to the consumer.
member_id	String	Consumer ID.
client_id	String	Client ID.

Table 5-158 assignment

Parameter	Type	Description
topic	String	Topic name.
partitions	Array of integers	Partition list.

Table 5-159 group_message_offsets

Parameter	Type	Description
partition	Integer	Partition number.
lag	Long	Number of remaining messages that can be retrieved, that is, the number of accumulated messages.
topic	String	Topic name.
message_current_offset	Long	Consumer offset.
message_log_end_offset	Long	Log end offset (LEO).

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/groups/{group}
```

Example Responses

Status code: 200

The consumer group details are queried successfully.

```
{
  "group": {
    "members": [ {
      "host": "/172.31.1.102",
      "assignment": [ {
        "topic": "test",
        "partitions": [ 0, 1, 2 ]
      }
    ]
  }
}
```

```
    }],  
    "member_id" : "consumer-1-6b8ee551-d499-47d4-9beb-ba1527496785",  
    "client_id" : "consumer-1"  
  }],  
  "state" : "STABLE",  
  "group_id" : "test-consumer-group",  
  "coordinator_id" : 2,  
  "group_message_offsets" : [ {  
    "partition" : 0,  
    "lag" : 31396,  
    "topic" : "test",  
    "message_current_offset" : 935,  
    "message_log_end_offset" : 32331  
  }, {  
    "partition" : 0,  
    "lag" : 0,  
    "topic" : "aaaa",  
    "message_current_offset" : 0,  
    "message_log_end_offset" : 0  
  }, {  
    "partition" : 1,  
    "lag" : 31279,  
    "topic" : "test",  
    "message_current_offset" : 1058,  
    "message_log_end_offset" : 32337  
  }, {  
    "partition" : 1,  
    "lag" : 0,  
    "topic" : "aaaa",  
    "message_current_offset" : 0,  
    "message_log_end_offset" : 0  
  }, {  
    "partition" : 2,  
    "lag" : 31603,  
    "topic" : "test",  
    "message_current_offset" : 739,  
    "message_log_end_offset" : 32342  
  } ],  
  "assignment_strategy" : "range"  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ShowGroupsSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowGroupsRequest request = new ShowGroupsRequest();
request.withInstanceId("{instance_id}");
request.withGroup("{group}");
try {
    ShowGroupsResponse response = client.showGroups(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowGroupsRequest()
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        response = client.show_groups(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowGroupsRequest{}
    request.InstanceId = "{instance_id}"
    request.Group = "{group}"
    response, err := client.ShowGroups(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The consumer group details are queried successfully.

Error Codes

See [Error Codes](#).

5.6.2 Querying All Consumer Groups

Function

This API is used to query all consumer groups.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/groups

Table 5-160 Path Parameters

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

Table 5-161 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	String	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	String	Maximum number of consumer group IDs returned in the current query. The default value is 10 . The value ranges from 1 to 50.
group	No	String	Filter consumer group names that contain specific keywords.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-162 Response body parameters

Parameter	Type	Description
groups	Array of GroupInfoSimple objects	All consumer groups.
total	Integer	Total number of consumer groups.

Table 5-163 GroupInfoSimple

Parameter	Type	Description
createdAt	Long	Creation time.
group_id	String	Consumer group ID.
state	String	Consumer group status. The value can be: <ul style="list-style-type: none"> • Dead: The consumer group has no members or metadata. • Empty: The consumer group has metadata but has no members. • PreparingRebalance: The consumer group is to be rebalanced. • CompletingRebalance: All members have joined the group. • Stable: Members in the consumer group can consume messages. "
coordinator_id	Integer	Coordinator ID.
group_desc	String	Consumer group description.
lag	Long	Number of accumulated messages.

Example Requests

Querying the consumer group list

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/groups?
offset={offset}&limit={limit}&group={group}
```

Example Responses

Status code: 200

All consumer groups are queried successfully.

```
{
  "groups" : [ {
```

```
"createdAt" : 1691401194847,
"group_id" : "consumer-1",
"state" : "EMPTY",
"coordinator_id" : 1,
"lag" : 0,
"group_desc" : null
}, {
"createdAt" : 1691401194960,
"group_id" : "consumer-2",
"state" : "STABLE",
"coordinator_id" : 2,
"lag" : 0,
"group_desc" : null
}, {
"createdAt" : 1691401207309,
"group_id" : "consumer-3",
"state" : "STABLE",
"coordinator_id" : 3,
"lag" : 0,
"group_desc" : null
} ],
"total" : 3
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListInstanceConsumerGroupsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstanceConsumerGroupsRequest request = new ListInstanceConsumerGroupsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListInstanceConsumerGroupsResponse response = client.listInstanceConsumerGroups(request);
            System.out.println(response.toString());
        }
    }
}
```

```
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListInstanceConsumerGroupsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_instance_consumer_groups(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListInstanceConsumerGroupsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListInstanceConsumerGroups(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	All consumer groups are queried successfully.

Error Codes

See [Error Codes](#).

5.6.3 Deleting Consumer Groups of a Kafka Instance in Batches

Function

This API is used to delete multiple consumer groups of a Kafka instance in batches.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/groups/batch-delete

Table 5-164 Path Parameters

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

Request Parameters

Table 5-165 Request body parameters

Parameter	Mandatory	Type	Description
group_ids	Yes	Array of strings	IDs of all consumer groups to be deleted.

Response Parameters

Status code: 200

Table 5-166 Response body parameters

Parameter	Type	Description
failed_groups	Array of failed_groups objects	List of consumer groups that failed to be deleted.
total	Integer	Number of records that fail to be deleted.

Table 5-167 failed_groups

Parameter	Type	Description
group_id	String	ID of consumer groups that failed to be deleted.
error_message	String	Cause of the deletion failure.

Example Requests

Batch deleting consumer groups

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/groups/batch-delete
{
  "group_ids" : [ "get-sync-group0", "get-sync-group1" ]
}
```

Example Responses

Status code: 200

The consumer groups are deleted successfully.

```
{
  "failed_groups" : [ {
    "group_id" : "test-1",
    "error_message" : "UNKNOW"
  }, {
    "group_id" : "test-2",
    "error_message" : "UNKNOW"
  } ],
  "total" : 2
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Batch deleting consumer groups

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
BatchDeleteGroupRequest request = new BatchDeleteGroupRequest();
request.withInstanceId("{instance_id}");
BatchDeleteGroupReq body = new BatchDeleteGroupReq();
List<String> listbodyGroupIds = new ArrayList<>();
listbodyGroupIds.add("get-sync-group0");
listbodyGroupIds.add("get-sync-group1");
body.withGroupIds(listbodyGroupIds);
request.withBody(body);
try {
    BatchDeleteGroupResponse response = client.batchDeleteGroup(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Batch deleting consumer groups

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteGroupRequest()
        request.instance_id = "{instance_id}"
        listGroupIdsbody = [
            "get-sync-group0",
            "get-sync-group1"
        ]
        request.body = BatchDeleteGroupReq(
            group_ids=listGroupIdsbody
        )
        response = client.batch_delete_group(request)
```



```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Batch deleting consumer groups

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteGroupRequest{}
    request.InstanceId = "{instance_id}"
    var listGroupIdsbody = []string{
        "get-sync-group0",
        "get-sync-group1",
    }
    request.Body = &model.BatchDeleteGroupReq{
        GroupIds: listGroupIdsbody,
    }
    response, err := client.BatchDeleteGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The consumer groups are deleted successfully.

Error Codes

See [Error Codes](#).

5.6.4 Creating a Consumer Group

Function

This API is used to create a consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/group

Table 5-168 Path Parameters

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

Request Parameters

Table 5-169 Request body parameters

Parameter	Mandatory	Type	Description
group_name	Yes	String	Consumer group name.
group_desc	No	String	Consumer group description.

Response Parameters

Status code: 200

Table 5-170 Response body parameters

Parameter	Type	Description
-	String	Specified IPv4 private IP addresses. The number of specified IP addresses must be less than or equal to the number of new brokers. If the number of specified IP addresses is less than the number of brokers, the unspecified brokers are randomly assigned private IP addresses.

Status code: 400

Table 5-171 Response body parameters

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

Example Requests

Creating a consumer group named test

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/group
{
  "group_name" : "test"
}
```

Example Responses

Status code: 200

Creation succeeded.

```
success
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a consumer group named test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateKafkaConsumerGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateKafkaConsumerGroupRequest request = new CreateKafkaConsumerGroupRequest();
        request.withInstanceId("{instance_id}");
        CreateGroupReq body = new CreateGroupReq();
        body.withGroupName("test");
        request.withBody(body);
        try {
            CreateKafkaConsumerGroupResponse response = client.createKafkaConsumerGroup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Creating a consumer group named test

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
```

```
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateKafkaConsumerGroupRequest()
    request.instance_id = "{instance_id}"
    request.body = CreateGroupReq(
        group_name="test"
    )
    response = client.create_kafka_consumer_group(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Creating a consumer group named test

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateKafkaConsumerGroupRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.CreateGroupReq{
        GroupName: "test",
    }
    response, err := client.CreateKafkaConsumerGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```

        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Creation succeeded.
400	Creation failed.

Error Codes

See [Error Codes](#).

5.6.5 Resetting Consumer Group Offset to the Specified Position

Function

Kafka instances do not support resetting the consumer offset online. Before resetting, stop the client for which the offset is to be reset. After a client is stopped, the server considers the client offline only after the time period specified in `ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG` (1000 ms by default).

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset

Table 5-172 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID.
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group name.

Request Parameters

Table 5-173 Request body parameters

Parameter	Mandatory	Type	Description
topic	No	String	Topic name.
partition	Yes	Integer	Partition number. The default value is -1 , indicating that all partitions are reset.
message_offset	No	Long	Resetting consumer group offset to the specified position. <ul style="list-style-type: none"> If this position is earlier than the current earliest offset, the offset will be reset to the earliest offset. If this offset is later than the current largest offset, the offset will be reset to the latest offset. Either message_offset or timestamp must be specified.
timestamp	No	Long	Specified time that the offset is to be reset to. The value is a Unix timestamp, in millisecond. <ul style="list-style-type: none"> If this time is earlier than the current earliest timestamp, the offset will be reset to the earliest timestamp. If this time is later than the current largest timestamp, the offset will be reset to the latest timestamp. Either message_offset or timestamp must be specified.

Response Parameters

None

Example Requests

- Resetting consumer group offset to the specified position

```
POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset
```

```
{
  "topic" : "test",
  "partition" : 0,
  "message_offset" : 10
}
```

- Resetting consumer group offset to the specified time

POST https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/groups/{group}/reset-message-offset

```
{
  "topic" : "test",
  "partition" : 0,
  "timestamp" : 1571812144000
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

- Resetting consumer group offset to the specified position

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetWithEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetMessageOffsetWithEngineRequest request = new ResetMessageOffsetWithEngineRequest();
        request.withInstanceId("{instance_id}");
        request.withGroup("{group}");
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();
```



```
body.withMessageOffset(10L);
body.withPartition(0);
body.withTopic("test");
request.withBody(body);
try {
    ResetMessageOffsetWithEngineResponse response =
client.resetMessageOffsetWithEngine(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- **Resetting consumer group offset to the specified time**

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetWithEngineSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetMessageOffsetWithEngineRequest request = new ResetMessageOffsetWithEngineRequest();
        request.withInstanceId("{instance_id}");
        request.withGroup("{group}");
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();
        body.withTimestamp(1571812144000L);
        body.withPartition(0);
        body.withTopic("test");
        request.withBody(body);
        try {
            ResetMessageOffsetWithEngineResponse response =
client.resetMessageOffsetWithEngine(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

- Resetting consumer group offset to the specified position

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetWithEngineRequest()
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        request.body = ResetMessageOffsetReq(
            message_offset=10,
            partition=0,
            topic="test"
        )
        response = client.reset_message_offset_with_engine(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- Resetting consumer group offset to the specified time

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before
running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ResetMessageOffsetWithEngineRequest()
    request.instance_id = "{instance_id}"
    request.group = "{group}"
    request.body = ResetMessageOffsetReq(
        timestamp=1571812144000,
        partition=0,
        topic="test"
    )
    response = client.reset_message_offset_with_engine(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- Resetting consumer group offset to the specified position

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.ResetMessageOffsetWithEngineRequest{}
request.InstanceId = "{instance_id}"
request.Group = "{group}"
messageOffsetResetMessageOffsetReq:= int64(10)
topicResetMessageOffsetReq:= "test"
request.Body = &model.ResetMessageOffsetReq{
    MessageOffset: &messageOffsetResetMessageOffsetReq,
    Partition: int32(0),
    Topic: &topicResetMessageOffsetReq,
}
response, err := client.ResetMessageOffsetWithEngine(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Resetting consumer group offset to the specified time

package main

```

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetWithEngineRequest{}
    request.InstanceId = "{instance_id}"
    request.Group = "{group}"
    timestampResetMessageOffsetReq:= int64(1571812144000)
    topicResetMessageOffsetReq:= "test"
    request.Body = &model.ResetMessageOffsetReq{
        Timestamp: &timestampResetMessageOffsetReq,
        Partition: int32(0),
        Topic: &topicResetMessageOffsetReq,
    }
    response, err := client.ResetMessageOffsetWithEngine(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Successful

Error Codes

See [Error Codes](#).

5.6.6 Querying the Offset of a Consumer Group

Function

This API is used to query the offset of a consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/message-offset

Table 5-174 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group name.

Table 5-175 Query Parameters

Parameter	Mandatory	Type	Description
topic	No	String	Topic name.

Parameter	Mandatory	Type	Description
partition	No	String	Partition name.
offset	No	String	Offset.
limit	No	String	Maximum value.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-176 Response body parameters

Parameter	Type	Description
group_message_of_fsets	Array of GroupMessageOfFsetsDetailEntity objects	Consumer group offset details.
total	Integer	Total.

Table 5-177 GroupMessageOffsetsDetailEntity

Parameter	Type	Description
partition	String	Partition.
message_current_offset	String	Current offset.
message_log_start_offset	Integer	Start offset.
message_log_end_offset	Integer	End offset.
consumer_id	String	Consumer ID.
host	String	Host name.
client_id	String	Client ID.

Example Requests

Querying the offset of a consumer group

GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/message-offset

Example Responses

Status code: 200

Successful

```
{
  "group_message_offsets" : [ ],
  "total" : 0
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.7 Modifying All Consumer Groups

Function

This API is used to modify all consumer groups.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{engine}/{project_id}/instances/{instance_id}/groups

Table 5-178 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-179 Request body parameters

Parameter	Mandatory	Type	Description
group_name	No	String	Consumer group name.
group_desc	No	String	Consumer group description.

Response Parameters

Status code: 200

Table 5-180 Response body parameters

Parameter	Type	Description
job_id	String	Background task ID.

Example Requests

```
PUT https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups
```

Example Responses

Status code: 200

Successful

```
{  
  "job_id" : "8a2c259182ab0e9d0182ab1882560010"  
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.8 Querying a Specified Consumer Group

Function

This API is used to query a specified consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Table 5-181 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Filter query by consumer group name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-182 Response body parameters

Parameter	Type	Description
group	group object	Consumer group information.

Table 5-183 group

Parameter	Type	Description
group_id	String	Consumer group name.

Parameter	Type	Description
state	String	Consumer group status. The value can be: <ul style="list-style-type: none"> • Dead: The consumer group has no members and no metadata. • Empty: The consumer group has metadata but has no members. • PreparingRebalance: The consumer group is to be rebalanced. • CompletingRebalance: All members have jointed the group. • Stable: Members in the consumer group can consume messages normally.
coordinator_id	Integer	Coordinator ID.
members	Array of members objects	Consumer list.
group_message_offsets	Array of group_message_offsets objects	Consumer offset.
assignment_strategy	String	Partition assignment policy.

Table 5-184 members

Parameter	Type	Description
host	String	Consumer address.
member_id	String	Consumer ID.
client_id	String	Client ID.

Table 5-185 group_message_offsets

Parameter	Type	Description
partition	Integer	Partition number.
lag	Long	Number of remaining messages that can be retrieved, that is, the number of accumulated messages.
topic	String	Topic name.

Parameter	Type	Description
message_current_offset	Long	Consumer offset.
message_log_end_offset	Long	Log end offset (LEO).

Example Requests

GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Example Responses

Status code: 200

Successful

```
{
  "group": null,
  "members": [ {
    "host": "/172.31.1.102",
    "member_id": "consumer-1-6b8ee551-d499-47d4-9beb-ba1527496785",
    "client_id": "consumer-1"
  } ],
  "state": "STABLE",
  "group_id": "test-consumer-group",
  "coordinator_id": 2,
  "group_message_offsets": [ {
    "partition": 0,
    "lag": 31396,
    "topic": "test",
    "message_current_offset": 935,
    "message_log_end_offset": 32331
  }, {
    "partition": 0,
    "lag": 0,
    "topic": "aaaa",
    "message_current_offset": 0,
    "message_log_end_offset": 0
  }, {
    "partition": 1,
    "lag": 31279,
    "topic": "test",
    "message_current_offset": 1058,
    "message_log_end_offset": 32337
  }, {
    "partition": 1,
    "lag": 0,
    "topic": "aaaa",
    "message_current_offset": 0,
    "message_log_end_offset": 0
  }, {
    "partition": 2,
    "lag": 31603,
    "topic": "test",
    "message_current_offset": 739,
    "message_log_end_offset": 32342
  } ],
  "assignment_strategy": "range"
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.9 Deleting a Specified Consumer Group

Function

This API is used to delete a specified consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Table 5-186 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group ID.

Request Parameters

None

Response Parameters

None

Example Requests

DELETE https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Example Responses

None

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.10 Modifying a Specified Consumer Group

Function

This API is used to modify a specified consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}

Table 5-187 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group ID.

Request Parameters

Table 5-188 Request body parameters

Parameter	Mandatory	Type	Description
group_name	No	String	Consumer group name.
group_desc	No	String	Consumer group description.

Response Parameters

None

Example Requests

```
PUT https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceConsumerGroupSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        UpdateInstanceConsumerGroupRequest request = new UpdateInstanceConsumerGroupRequest();
        request.withEngine("{engine}");
        request.withInstanceId("{instance_id}");
        request.withGroup("{group}");
        CreateGroupReq body = new CreateGroupReq();
        request.withBody(body);
        try {
            UpdateInstanceConsumerGroupResponse response = client.updateInstanceConsumerGroup(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceConsumerGroupRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        request.group = "{group}"
        request.body = CreateGroupReq(
        )
        response = client.update_instance_consumer_group(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceConsumerGroupRequest{}
    request.Engine = "{engine}"
    request.InstanceId = "{instance_id}"
    request.Group = "{group}"
    request.Body = &model.CreateGroupReq{
    }
    response, err := client.UpdateInstanceConsumerGroup(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.11 Querying Topics of a Specified Consumer Group

Function

This API is used to query topics of a specified consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/topics

Table 5-189 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group ID.

Table 5-190 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	Integer	Maximum number of topics returned in the current query. The default value is 10 . The value ranges from 1 to 50.
sort_key	No	String	Sort by: <ul style="list-style-type: none"> • topic: topic name • partition: number of partitions • messages (default): number of messages

Parameter	Mandatory	Type	Description
sort_dir	No	String	Sort in: <ul style="list-style-type: none"> • asc: ascending order • desc (default): descending order
topic	No	String	Topic name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-191 Response body parameters

Parameter	Type	Description
topics	Array of GroupTopicEntity objects	Consumer group topic.
total	Integer	Total.

Table 5-192 GroupTopicEntity

Parameter	Type	Description
topic	String	Topic name.
partitions	Integer	Partition.
lag	Integer	Number of stacked messages.

Example Requests

```
GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/topics
```

Example Responses

Status code: 200

Successful

```
{
  "topics": [ {
    "topic": "topic-1",
    "partitions": 1,
```

```

    "lag" : 0
  }, {
    "topic" : "topic-2",
    "partitions" : 2,
    "lag" : 10
  } ],
  "total" : 2
}

```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.6.12 Querying Consumers in a Specified Consumer Group

Function

This API is used to query consumers in a specified consumer group.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/members

Table 5-193 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group ID.

Table 5-194 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	Integer	Maximum number of consumers returned in the current query. The default value is 10 . The value ranges from 1 to 50.
host	No	String	Consumer address.
member_id	No	String	Consumer ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-195 Response body parameters

Parameter	Type	Description
members	Array of GroupMemberEntity objects	Consumer details.
total	Integer	Total.

Table 5-196 GroupMemberEntity

Parameter	Type	Description
member_id	String	Consumer ID.
client_id	String	Client ID.

Example Requests

GET `https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups/{group}/members`

Example Responses

Status code: 200

Successful

```
{
  "members" : [ {
    "member_id" : "consumer-1-6b8ee551-d499-47d4-9beb-ba1527496785",
    "client_id" : "consumer-1"
  }, {
    "member_id" : "consumer-2-6b8ee551-d499-47d4-9beb-ba1527491125",
    "client_id" : "consumer-2"
  } ],
  "total" : 2
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.7 User Management

5.7.1 Querying the User List

Function

This API is used to query the user list.

User management is supported only when SASL is enabled for the Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/users

Table 5-197 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-198 Response body parameters

Parameter	Type	Description
users	Array of ShowInstanceUsersEntity objects	User list.

Table 5-199 ShowInstanceUsersEntity

Parameter	Type	Description
user_name	String	Username. A username must start with a letter. It can contain 4 to 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.
user_desc	String	User description.
role	String	User role.
default_app	Boolean	Whether an application is the default application.
created_time	Long	Creation time.

Example Requests

Querying the user list.

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
```

Example Responses

Status code: 200

The query is successful.

```
{
  "users" : [ {
    "user_name" : "xxxa",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1615431764734
  }, {
    "user_name" : "test",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1615364062463
  }, {
    "user_name" : "ROOT",
    "role" : "guest",
    "default_app" : false,
    "created_time" : 1617194246328
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceUsersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowInstanceUsersRequest request = new ShowInstanceUsersRequest();
        request.withInstanceId("{instance_id}");
        try {
```

```
        ShowInstanceUsersResponse response = client.showInstanceUsers(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceUsersRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_users(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```



```
variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowInstanceUsersRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowInstanceUsers(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.7.2 Creating a User

Function

This API is used to create a user for a Kafka instance for which SASL is enabled. For Kafka instances created before July 15, 2023, a maximum of 20 users can be created for each instance. For Kafka instances created since July 15, 2023, a maximum of 500 users can be created for each instance.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/instances/{instance_id}/users

Table 5-200 Path Parameters

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

Request Parameters

Table 5-201 Request body parameters

Parameter	Mandatory	Type	Description
user_name	No	String	Username. This parameter is mandatory for creating a user.
user_desc	No	String	User description.
user_passwd	No	String	Password. This parameter is mandatory for creating a user. The password must be different from the username. The password must meet the following complexity requirements: <ul style="list-style-type: none"> • Can contain 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters include (~!@#\$%^&*()-_+= []{}:~'"<.>/?) and spaces, and cannot start with a hyphen (-).

Response Parameters

Status code: 400

Table 5-202 Response body parameters

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

Status code: 403

Table 5-203 Response body parameters

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

Example Requests

Creating a user whose username is test and password is Cxxx3

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
{
  "user_name" : "test",
  "user_passwd" : "Cxxx3"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a user whose username is test and password is Cxxx3

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class CreateInstanceUserSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateInstanceUserRequest request = new CreateInstanceUserRequest();
        request.withInstanceld("{instance_id}");
        CreateInstanceUserReq body = new CreateInstanceUserReq();
        body.withUserPasswd("Cxxx3");
        body.withUserName("test");
        request.withBody(body);
        try {
            CreateInstanceUserResponse response = client.createInstanceUser(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Creating a user whose username is test and password is Cxxx3

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateInstanceUserRequest()
    request.instance_id = "{instance_id}"
    request.body = CreateInstanceUserReq(
        user_passwd="Cxxx3",
        user_name="test"
    )
    response = client.create_instance_user(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Creating a user whose username is test and password is Cxxx3

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateInstanceUserRequest{}
    request.InstanceId = "{instance_id}"
    userPasswdCreateInstanceUserReq := "Cxxx3"
    userNameCreateInstanceUserReq := "test"
    request.Body = &model.CreateInstanceUserReq{
        UserPasswd: &userPasswdCreateInstanceUserReq,
        UserName: &userNameCreateInstanceUserReq,
    }
    response, err := client.CreateInstanceUser(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```

        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The creation is successful.
400	Invalid parameters.
403	Authentication failed.

Error Codes

See [Error Codes](#).

5.7.3 Deleting Users in Batches

Function

This API is used to delete multiple users of a Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/users

Table 5-204 Path Parameters

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

Request Parameters

Table 5-205 Request body parameters

Parameter	Mandatory	Type	Description
action	No	String	Deletion type. Currently, only delete is supported. This parameter is mandatory for deleting a user.
users	No	Array of strings	User list. This parameter is mandatory for deleting a user.

Response Parameters

None

Example Requests

Deleting users in batches.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/users
{
  "action": "delete",
  "users": [ "testuser" ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Deleting users in batches.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;
```

```
public class BatchDeleteInstanceUsersSolution {
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();

        BatchDeleteInstanceUsersRequest request = new BatchDeleteInstanceUsersRequest();
        request.withInstanceId("{instance_id}");
        BatchDeleteInstanceUsersReq body = new BatchDeleteInstanceUsersReq();
        List<String> listbodyUsers = new ArrayList<>();
        listbodyUsers.add("testuser");
        body.withUsers(listbodyUsers);
        body.withAction(BatchDeleteInstanceUsersReq.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchDeleteInstanceUsersResponse response = client.batchDeleteInstanceUsers(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Deleting users in batches.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```



```
client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchDeleteInstanceUsersRequest()
    request.instance_id = "{instance_id}"
    listUsersbody = [
        "testuser"
    ]
    request.body = BatchDeleteInstanceUsersReq(
        users=listUsersbody,
        action="delete"
    )
    response = client.batch_delete_instance_users(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Deleting users in batches.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchDeleteInstanceUsersRequest{}
    request.InstanceId = "{instance_id}"
    var listUsersbody = []string{
        "testuser",
    }
    }
    actionBatchDeleteInstanceUsersReq:= model.GetBatchDeleteInstanceUsersReqActionEnum().DELETE
    request.Body = &model.BatchDeleteInstanceUsersReq{
        Users: &listUsersbody,
        Action: &actionBatchDeleteInstanceUsersReq,
```

```

}
response, err := client.BatchDeleteInstanceUsers(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The deletion is successful.

Error Codes

See [Error Codes](#).

5.7.4 Resetting a User Password

Function

This API is used to reset a user password.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{project_id}/instances/{instance_id}/users/{user_name}

Table 5-206 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
user_name	Yes	String	Username.

Request Parameters

Table 5-207 Request body parameters

Parameter	Mandatory	Type	Description
new_password	No	String	<p>New password.</p> <p>This parameter is mandatory for resetting a user password. It cannot be the username or the username spelled backwards.</p> <p>The password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Can contain 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters include (~!@#\$%^&*()-_+= []{}:~<.>/?) and spaces, and cannot start with a hyphen (-).

Response Parameters

None

Example Requests

Resetting a user password.

```
PUT https://{endpoint}/v2/{project_id}/instances/{instance_id}/users/{user_name}
{
  "new_password" : "Cxxx3"
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Resetting a user password.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetUserPasswrodSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetUserPasswrodRequest request = new ResetUserPasswrodRequest();
        request.withInstanceId("{instance_id}");
        request.withUserName("{user_name}");
        ResetUserPasswrodReq body = new ResetUserPasswrodReq();
        body.withNewPassword("Cxxx3");
        request.withBody(body);
        try {
            ResetUserPasswrodResponse response = client.resetUserPasswrod(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Resetting a user password.

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetUserPasswrodRequest()
        request.instance_id = "{instance_id}"
        request.user_name = "{user_name}"
        request.body = ResetUserPasswrodReq(
            new_password="Cxxx3"
        )
        response = client.reset_user_passwrod(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Resetting a user password.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

        request := &model.ResetUserPasswrodRequest{}
        request.InstanceId = "{instance_id}"
        request.UserName = "{user_name}"
        newPasswordResetUserPasswrodReq:= "Cxxx3"
        request.Body = &model.ResetUserPasswrodReq{
            NewPassword: &newPasswordResetUserPasswrodReq,
        }
        response, err := client.ResetUserPasswrod(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Password reset successfully.

Error Codes

See [Error Codes](#).

5.7.5 Modifying User Parameters

Function

This API is used to modify user parameters.

Calling Method

For details, see [Calling APIs](#).

URI

PUT /v2/{engine}/{project_id}/instances/{instance_id}/users/{user_name}

Table 5-208 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
user_name	Yes	String	Username.

Request Parameters

Table 5-209 Request body parameters

Parameter	Mandatory	Type	Description
new_password	No	String	<p>New password.</p> <p>It cannot be the username or the username spelled backwards.</p> <p>The password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Can contain 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters include (~!@#\$%^&*()-_+ [{}]:'"<.>/?) and spaces, and cannot start with a hyphen (-).
user_name	No	String	Username.
user_desc	No	String	User description.

Response Parameters

None

Example Requests

Modifying user parameters

```
PUT https://{endpoint}/v2/kafka/{project_id}/instances/{instance_id}/users/{user_name}
{
  "new_password" : "Cxxx3",
  "user_name" : "Username.",
  "user_desc" : "User description."
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying user parameters

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateInstanceUserSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceUserRequest request = new UpdateInstanceUserRequest();
        request.withEngine("{engine}");
        request.withInstanceId("{instance_id}");
        request.withUserName("{user_name}");
        UpdateUserReq body = new UpdateUserReq();
        body.withUserDesc("User description.");
        body.withUserName("Username.");
        body.withNewPassword("Cxxx3");
        request.withBody(body);
        try {
            UpdateInstanceUserResponse response = client.updateInstanceUser(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```



```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Modifying user parameters

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateInstanceUserRequest()
        request.engine = "{engine}"
        request.instance_id = "{instance_id}"
        request.user_name = "{user_name}"
        request.body = UpdateUserReq(
            user_desc="User description.",
            user_name="Username.",
            new_password="Cxxx3"
        )
        response = client.update_instance_user(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Modifying user parameters

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceUserRequest{}
    request.Engine = "{engine}"
    request.InstanceId = "{instance_id}"
    request.UserName = "{user_name}"
    userDescUpdateUserReq:= "User description."
    userNameUpdateUserReq:= "Username."
    newPasswordUpdateUserReq:= "Cxxx3"
    request.Body = &model.UpdateUserReq{
        UserDesc: &userDescUpdateUserReq,
        UserName: &userNameUpdateUserReq,
        NewPassword: &newPasswordUpdateUserReq,
    }
    response, err := client.UpdateInstanceUser(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Modified.

Error Codes

See [Error Codes](#).

5.7.6 Querying User Permissions

Function

This API is used to query user permissions.

User management is supported only when SASL is enabled for the Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy

Table 5-210 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic_name	Yes	String	Topic name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-211 Response body parameters

Parameter	Type	Description
name	String	Consumer group description.
topic_type	Integer	Specified IPv4 private IP addresses. The number of specified IP addresses must be less than or equal to the number of new brokers. If the number of specified IP addresses is less than the number of brokers, the unspecified brokers are randomly assigned private IP addresses.

Parameter	Type	Description
policies	Array of PolicyEntity objects	New brokers use the standby subnet ID in instance expansion. This value is transferred when a standby subnet is used in instance expansion. Contact customer service to use the value.

Table 5-212 PolicyEntity

Parameter	Type	Description
owner	Boolean	Whether the user is the one selected during topic creation.
user_name	String	Username.
access_policy	String	Permission type. <ul style="list-style-type: none"> • all: publish and subscribe permissions. • pub: publish permissions. • sub: subscribe permissions.

Example Requests

Querying user permissions for a topic.

```
GET https://{endpoint}/v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy
```

Example Responses

Status code: 200

The query is successful.

```
{
  "name" : "topic-test",
  "policies" : [ {
    "owner" : false,
    "user_name" : "xxa",
    "access_policy" : "pub"
  }, {
    "owner" : false,
    "user_name" : "root",
    "access_policy" : "all"
  } ],
  "topic_type" : 0
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowTopicAccessPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTopicAccessPolicyRequest request = new ShowTopicAccessPolicyRequest();
        request.withInstanceId("{instance_id}");
        request.withTopicName("{topic_name}");
        try {
            ShowTopicAccessPolicyResponse response = client.showTopicAccessPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTopicAccessPolicyRequest()
        request.instance_id = "{instance_id}"
        request.topic_name = "{topic_name}"
        response = client.show_topic_access_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTopicAccessPolicyRequest{}
    request.InstanceId = "{instance_id}"
    request.TopicName = "{topic_name}"
    response, err := client.ShowTopicAccessPolicy(request)
    if err == nil {
```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.7.7 Granting User Permissions

Function

This API is used to grant user permissions.

User management is supported only when SASL is enabled for the Kafka instance.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v1/{project_id}/instances/{instance_id}/topics/accesspolicy

Table 5-213 Path Parameters

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

Request Parameters

Table 5-214 Request body parameters

Parameter	Mandatory	Type	Description
topics	Yes	Array of AccessPolicyTopicEntity objects	Topic list.

Table 5-215 AccessPolicyTopicEntity

Parameter	Mandatory	Type	Description
name	Yes	String	Topic name.
policies	Yes	Array of AccessPolicyEntity objects	Permission list.

Table 5-216 AccessPolicyEntity

Parameter	Mandatory	Type	Description
user_name	No	String	Username. This parameter is mandatory when you set user permissions.
access_policy	No	String	Permission type. <ul style="list-style-type: none"> • all: publish and subscribe permissions. • pub: publish permissions. • sub: subscribe permissions. This parameter is mandatory when you set user permissions.

Response Parameters

Status code: 400

Table 5-217 Response body parameters

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

Status code: 403

Table 5-218 Response body parameters

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

Example Requests

Granting the root user the permission to publish and subscribe to topic-test

POST https://{endpoint}/v1/{project_id}/instances/{instance_id}/topics/accesspolicy

```
{
  "topics" : [ {
    "name" : "topic-test",
    "policies" : [ {
      "user_name" : "root",
      "access_policy" : "all"
    } ]
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Granting the root user the permission to publish and subscribe to topic-test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
import java.util.List;
import java.util.ArrayList;

public class UpdateTopicAccessPolicySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTopicAccessPolicyRequest request = new UpdateTopicAccessPolicyRequest();
        request.withInstanceId("{instance_id}");
        UpdateTopicAccessPolicyReq body = new UpdateTopicAccessPolicyReq();
        List<AccessPolicyEntity> listTopicsPolicies = new ArrayList<>();
        listTopicsPolicies.add(
            new AccessPolicyEntity()
                .withUserName("root")
                .withAccessPolicy(AccessPolicyEntity.AccessPolicyEnum.fromValue("all"))
        );
        List<AccessPolicyTopicEntity> listbodyTopics = new ArrayList<>();
        listbodyTopics.add(
            new AccessPolicyTopicEntity()
                .withName("topic-test")
                .withPolicies(listTopicsPolicies)
        );
        body.withTopics(listbodyTopics);
        request.withBody(body);
        try {
            UpdateTopicAccessPolicyResponse response = client.updateTopicAccessPolicy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Granting the root user the permission to publish and subscribe to topic-test

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTopicAccessPolicyRequest()
        request.instance_id = "{instance_id}"
        listPoliciesTopics = [
            AccessPolicyEntity(
                user_name="root",
                access_policy="all"
            )
        ]
        listTopicsbody = [
            AccessPolicyTopicEntity(
                name="topic-test",
                policies=listPoliciesTopics
            )
        ]
        request.body = UpdateTopicAccessPolicyReq(
            topics=listTopicsbody
        )
        response = client.update_topic_access_policy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Granting the root user the permission to publish and subscribe to topic-test

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"
```

```

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateTopicAccessPolicyRequest{
    request.InstanceId = "{instance_id}"
    userNamePolicies:= "root"
    accessPolicyPolicies:= model.GetAccessPolicyEntityAccessPolicyEnum().ALL
    var listPoliciesTopics = []model.AccessPolicyEntity{
        {
            UserName: &userNamePolicies,
            AccessPolicy: &accessPolicyPolicies,
        },
    }
    var listTopicsbody = []model.AccessPolicyTopicEntity{
        {
            Name: "topic-test",
            Policies: listPoliciesTopics,
        },
    }
    request.Body = &model.UpdateTopicAccessPolicyReq{
        Topics: listTopicsbody,
    }
    response, err := client.UpdateTopicAccessPolicy(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The update is successful.
400	Invalid parameters.
403	Authentication failed.

Error Codes

See [Error Codes](#).

5.8 Managing Messages

5.8.1 Querying Messages

Function

This API is used to query the offset and content of a message.

This API queries the message offset based on the timestamp and then queries the message content based on the offset.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/messages

Table 5-219 Path Parameters

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

Table 5-220 Query Parameters

Parameter	Mandatory	Type	Description
topic	Yes	String	Topic name. A topic name must start with a letter and can only contain letters, hyphens (-), underscores (_), and digits.
asc	No	Boolean	Whether to sort messages by time.
start_time	No	String	Start time. The value is a Unix timestamp, in millisecond. This parameter is mandatory when you query the message offset.

Parameter	Mandatory	Type	Description
end_time	No	String	End time. The value is a Unix timestamp, in millisecond. This parameter is mandatory when you query the message offset.
limit	No	String	Number of messages displayed on each page.
offset	No	String	Page number.
download	No	Boolean	Whether download is required.
message_offset	No	String	Message offset. This parameter is mandatory when you query the message content. If start_time and end_time are not empty, this parameter is invalid.
partition	No	String	Partition. This parameter is mandatory when you query the message content. If start_time and end_time are not empty, this parameter is invalid.
keyword	No	String	Keyword. The value ranges from 0 to 50.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-221 Response body parameters

Parameter	Type	Description
messages	Array of MessagesEntity objects	Message list.

Parameter	Type	Description
total	Long	Total number of messages.
size	Long	Number of records on each page.

Table 5-222 MessagesEntity

Parameter	Type	Description
topic	String	Topic name.
partition	Integer	Partition where the message is located.
key	String	Message key.
value	String	Message content.
size	Integer	Message size.
timestamp	Long	Message production time. The value is a UNIX timestamp, in ms.
huge_message	Boolean	Big data flag.
message_offset	Long	Message offset.
message_id	String	Message ID.
app_id	String	Application ID.
tag	String	Message label.

Status code: 400

Table 5-223 Response body parameters

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

Status code: 403

Table 5-224 Response body parameters

Parameter	Type	Description
error_code	String	Error code.

Parameter	Type	Description
error_msg	String	Error description.

Example Requests

- Querying the message offset.

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages?asc=false&end_time=1608609032042&limit=10&offset=0&start_time=1608608432042&topic=topic-test
```

- Querying the message content.

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/messages?download=false&message_offset=0&partition=0&topic=topic-test
```

Example Responses

Status code: 200

The query is successful.

```
{
  "messages": [ {
    "topic": "topic-test",
    "partition": 0,
    "value": "hello world",
    "size": 21,
    "timestamp": 1607598463502,
    "huge_message": false,
    "message_offset": 4,
    "message_id": "",
    "app_id": "",
    "tag": ""
  } ],
  "total": 1,
  "size": 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowInstanceMessagesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
```



```
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowInstanceMessagesRequest request = new ShowInstanceMessagesRequest();
request.withInstanceId("{instance_id}");
try {
    ShowInstanceMessagesResponse response = client.showInstanceMessages(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowInstanceMessagesRequest()
        request.instance_id = "{instance_id}"
        response = client.show_instance_messages(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowInstanceMessagesRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowInstanceMessages(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.
400	Invalid parameters.
403	Authentication failed.

Error Codes

See [Error Codes](#).

5.8.2 Querying a Message with a Specified Offset

Function

This API is used to query a message with a specified offset.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/message

Table 5-225 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name. A topic name must start with a letter and can only contain letters, hyphens (-), underscores (_), and digits.
partition	Yes	Integer	Partition number.

Table 5-226 Query Parameters

Parameter	Mandatory	Type	Description
message_offset	Yes	String	Message offset.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-227 Response body parameters

Parameter	Type	Description
message	Array of ShowPartitionMessageEntity objects	Message list.

Table 5-228 ShowPartitionMessageEntity

Parameter	Type	Description
key	String	Message key.
value	String	Message content.
topic	String	Topic name.
partition	Integer	Partition number.
message_offset	Long	Message offset.
size	Integer	Message size in bytes.
timestamp	Long	Time when a message is created. The value is a Unix timestamp. The unit is millisecond.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/message?message_offset={message_offset}
```

Example Responses

Status code: 200

The message with the specified offset is queried successfully.

```
{
  "message": [ {
    "topic": "mytest",
    "partition": 0,
    "message_offset": 7,
    "key": null,
    "value": "kasjdf",
    "size": 6,
    "timestamp": 1568125036045
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowPartitionMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowPartitionMessageRequest request = new ShowPartitionMessageRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        request.withPartition("{partition}");
        try {
            ShowPartitionMessageResponse response = client.showPartitionMessage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowPartitionMessageRequest()
    request.instance_id = "{instance_id}"
    request.topic = "{topic}"
    request.partition = {partition}
    response = client.show_partition_message(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionMessageRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    request.Partition = int32({partition})
    response, err := client.ShowPartitionMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```

        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The message with the specified offset is queried successfully.

Error Codes

See [Error Codes](#).

5.8.3 Querying a Message with a Specified Time Period

Function

This API is used to query a message with a specified time period.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/messages

Table 5-229 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name. A topic name must start with a letter and can only contain letters, hyphens (-), underscores (_), and digits.

Table 5-230 Query Parameters

Parameter	Mandatory	Type	Description
start_time	No	String	Query start time as a Unix timestamp. Default value: 0 .
end_time	No	String	Query end time, as a Unix timestamp. Default value: current system time.
limit	No	Integer	Number of messages returned on a page. Default value: 10 .
offset	No	Integer	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
partition	No	String	Partition number. The default value is -1 , indicating that all partitions are queried.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-231 Response body parameters

Parameter	Type	Description
messages	Array of messages objects	Message list.
messages_count	Integer	Total number of messages.
offsets_count	Integer	Total number of pages.
offset	Integer	Current page.

Table 5-232 messages

Parameter	Type	Description
topic	String	Topic name.
partition	Integer	Partition number.

Parameter	Type	Description
message_offset	Long	Message offset.
size	Integer	Message size in bytes.
timestamp	Long	Time when a message is created. The value is a Unix timestamp. The unit is millisecond.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/messages
```

Example Responses

Status code: 200

The message with the specified time period is queried successfully.

```
{
  "messages": [ {
    "topic": "mytest",
    "partition": 0,
    "message_offset": 7,
    "size": 6,
    "timestamp": 1568125036045
  } ],
  "messages_count": 1,
  "offsets_count": 1,
  "offset": 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowMessagesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();

ShowMessagesRequest request = new ShowMessagesRequest();
request.withInstanceId("{instance_id}");
request.withTopic("{topic}");
try {
    ShowMessagesResponse response = client.showMessages(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMessagesRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        response = client.show_messages(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMessagesRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    response, err := client.ShowMessages(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The message with the specified time period is queried successfully.

Error Codes

See [Error Codes](#).

5.8.4 Querying Offset of the Earliest Message in a Partition

Function

This API is used to query the offset of the earliest message in a partition.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/beginning-message

Table 5-233 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name. A topic name must start with a letter and can only contain letters, hyphens (-), underscores (_), and digits.
partition	Yes	Integer	Partition number.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-234 Response body parameters

Parameter	Type	Description
topic	String	Topic name.
partition	Integer	Partition number.
offset	Integer	Offset of the latest message.

Parameter	Type	Description
timestamp	Long	Time when a message is created. The value is a Unix timestamp. The unit is millisecond.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/beginning-message
```

Example Responses

Status code: 200

The offset of the earliest message in a partition is queried successfully.

```
{
  "topic" : "mytest",
  "partition" : 0,
  "offset" : 9,
  "timestamp" : 1568125039164
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowPartitionBeginningMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        ShowPartitionBeginningMessageRequest request = new ShowPartitionBeginningMessageRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        request.withPartition({partition});
        try {
            ShowPartitionBeginningMessageResponse response =
client.showPartitionBeginningMessage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPartitionBeginningMessageRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        request.partition = {partition}
        response = client.show_partition_beginning_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionBeginningMessageRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    request.Partition = int32({partition})
    response, err := client.ShowPartitionBeginningMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The offset of the earliest message in a partition is queried successfully.

Error Codes

See [Error Codes](#).

5.8.5 Querying Offset of the Latest Message in a Partition

Function

This API is used to query the offset of the latest message in a partition.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/end-message

Table 5-235 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name. A topic name must start with a letter and can only contain letters, hyphens (-), underscores (_), and digits.
partition	Yes	Integer	Partition number.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-236 Response body parameters

Parameter	Type	Description
topic	String	Topic name.
partition	Integer	Partition number.
offset	Integer	Offset of the latest message.

Parameter	Type	Description
timestamp	Long	Time when a message is created. The value is a Unix timestamp. The unit is millisecond.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions/{partition}/end-message
```

Example Responses

Status code: 200

The offset of the latest message in a partition is queried successfully.

```
{
  "topic" : "mytest",
  "partition" : 0,
  "offset" : 9,
  "timestamp" : 1568125039164
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowPartitionEndMessageSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        ShowPartitionEndMessageRequest request = new ShowPartitionEndMessageRequest();
        request.withInstanceId("{instance_id}");
        request.withTopic("{topic}");
        request.withPartition({partition});
        try {
            ShowPartitionEndMessageResponse response = client.showPartitionEndMessage(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowPartitionEndMessageRequest()
        request.instance_id = "{instance_id}"
        request.topic = "{topic}"
        request.partition = {partition}
        response = client.show_partition_end_message(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```

kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPartitionEndMessageRequest{}
    request.InstanceId = "{instance_id}"
    request.Topic = "{topic}"
    request.Partition = int32({partition})
    response, err := client.ShowPartitionEndMessage(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The offset of the latest message in a partition is queried successfully.

Error Codes

See [Error Codes](#).

5.8.6 Deleting a Kafka Message

Function

This API is used to delete a Kafka message.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/messages

Table 5-237 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name.

Request Parameters

Table 5-238 Request body parameters

Parameter	Mandatory	Type	Description
partitions	No	Array of PartitionOffsetEntity objects	Partition offset details.

Table 5-239 PartitionOffsetEntity

Parameter	Mandatory	Type	Description
partition	No	Integer	Partition.
offset	No	Integer	Consumer offset.

Response Parameters

Status code: 200

Table 5-240 Response body parameters

Parameter	Type	Description
partitions	Array of PartitionResp objects	Partition response information.

Table 5-241 PartitionResp

Parameter	Type	Description
partition	Integer	Partition.
result	String	Returned result.
error_code	String	Returned error code.

Example Requests

```
DELETE https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/topics/{topic}/messages
```

Example Responses

Status code: 200

Successful

```
{
  "partitions": [ {
    "partition": 0,
    "result": "success",
    "error_code": 0
  } ]
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.9 Background Task Management

5.9.1 Listing Background Tasks

Function

This API is used to list background tasks of an instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/tasks

Table 5-242 Path Parameters

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

Table 5-243 Query Parameters

Parameter	Mandatory	Type	Description
start	No	Integer	ID of the task where the query starts.
limit	No	Integer	Number of tasks to be queried.
begin_time	No	String	Time of task where the query starts. The format is YYYYMMDDHHmmss.
end_time	No	String	Time of task where the query ends. The format is YYYYMMDDHHmmss.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-244 Response body parameters

Parameter	Type	Description
task_count	String	Number of tasks.
tasks	Array of tasks objects	Task list.

Table 5-245 tasks

Parameter	Type	Description
id	String	Task ID.
name	String	Task name.
user_name	String	Username.
user_id	String	User ID.
params	String	Task parameters.
status	String	Task status.
created_at	String	Start time.
updated_at	String	End time.

Example Requests

```
'GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks?
start={start}&limit={limit}&begin_time={begin_time}&end_time={end_time}'
```

Example Responses

Status code: 200

Background tasks are listed successfully.

```
{
  "task_count" : "4",
  "tasks" : [ {
    "id" : "8abfa7b372160bfd0172165864064079",
    "name" : "modifyAutoTopic",
    "user_name" : "paas_dms",
    "user_id" : "3df5acbc24a54fadb62a043c9000a307",
    "params" : "{\"old_auto_status\":true,\"new_auto_status\":false}",
    "status" : "EXECUTING",
    "created_at" : "2020-05-15T03:19:51.046Z",
    "updated_at" : "2020-05-15T03:19:51.065Z"
  }, {
    "id" : "8abfa7b372160bfd017216560af83e6e",
    "name" : "changeRetentionPolicy",
    "user_name" : "paas_dms",
    "user_id" : "3df5acbc24a54fadb62a043c9000a307",
    "params" : "{\"new_retention_policy\":\"produce_reject\",\"origin_retention_policy\":\"time_base\"}",
    "status" : "SUCCESS",
    "created_at" : "2020-05-15T03:17:17.176Z",
```

```
"updated_at" : "2020-05-15T03:17:22.162Z"  
}]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ListBackgroundTasksSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListBackgroundTasksRequest request = new ListBackgroundTasksRequest();  
        request.withInstanceId("{instance_id}");  
        try {  
            ListBackgroundTasksResponse response = client.listBackgroundTasks(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
import os
```



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBackgroundTasksRequest()
        request.instance_id = "{instance_id}"
        response = client.list_background_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBackgroundTasksRequest{}
    request.InstanceId = "{instance_id}"
```

```
response, err := client.ListBackgroundTasks(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Background tasks are listed successfully.

Error Codes

See [Error Codes](#).

5.9.2 Querying a Background Task

Function

This API is used to query a specified background task.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}

Table 5-246 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
task_id	Yes	String	Task ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-247 Response body parameters

Parameter	Type	Description
task_count	String	Number of tasks.
tasks	Array of tasks objects	Task list.

Table 5-248 tasks

Parameter	Type	Description
id	String	Task ID.
name	String	Task name.
user_name	String	Username.
user_id	String	User ID.
params	String	Task parameters.
status	String	Task status.
created_at	String	Start time.
updated_at	String	End time.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks/{task_id}
```

Example Responses

Status code: 200

The query is successful.

```
{
  "task_count": "1",
  "tasks": [ {
    "id": "8abfa7b272adc5b40172b73130065ae7",
    "name": "bindInstancePublicIp",
    "user_name": "paas_dms",
    "user_id": "3df5acbc24a54fadb62a043c9000a307",
    "params": "{\"public_ip_id\":\"1aea7aed-e7d8-40ea-b3de-6f3ee9d5db9f\",\"public_ip_address\":\"100.93.2.18\", \"enable_public_ip\":true}",
  } ]
}
```

```
"status" : "SUCCESS",  
"created_at" : "2020-06-15T08:55:53.606Z",  
"updated_at" : "2020-06-15T08:55:56.600Z"  
}]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ShowBackgroundTaskSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowBackgroundTaskRequest request = new ShowBackgroundTaskRequest();  
        request.withInstanceId("{instance_id}");  
        request.withTaskId("{task_id}");  
        try {  
            ShowBackgroundTaskResponse response = client.showBackgroundTask(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowBackgroundTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.show_background_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
```

```

        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build()

    request := &model.ShowBackgroundTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowBackgroundTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The query is successful.

Error Codes

See [Error Codes](#).

5.9.3 Deleting a Background Task

Function

This API is used to delete a specified background task.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}

Table 5-249 Path Parameters

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

Parameter	Mandatory	Type	Description
task_id	Yes	String	Task ID.

Request Parameters

None

Response Parameters

None

Example Requests

Deleting a specified background task

```
DELETE https://{endpoint}/v2/{project_id}/instances/{instance_id}/tasks/{task_id}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteBackgroundTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
        .build();
DeleteBackgroundTaskRequest request = new DeleteBackgroundTaskRequest();
request.withInstanceId("{instance_id}");
request.withTaskId("{task_id}");
try {
    DeleteBackgroundTaskResponse response = client.deleteBackgroundTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteBackgroundTaskRequest()
        request.instance_id = "{instance_id}"
        request.task_id = "{task_id}"
        response = client.delete_background_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
```



```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBackgroundTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.TaskId = "{task_id}"
    response, err := client.DeleteBackgroundTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	The background task is deleted successfully.

Error Codes

See [Error Codes](#).

5.10 Tag Management

5.10.1 Batch Adding or Deleting Tags

Function

This API is used to add or delete instance tags in batches.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/kafka/{instance_id}/tags/action

Table 5-250 Path Parameters

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

Request Parameters

Table 5-251 Request body parameters

Parameter	Mandatory	Type	Description
action	No	String	Operation. Only lowercase letters are supported. <ul style="list-style-type: none"> create: Tags are created. delete: Tags are deleted.
tags	No	Array of TagEntity objects	Tag list.

Table 5-252 TagEntity

Parameter	Mandatory	Type	Description
key	No	String	Tag key. <ul style="list-style-type: none"> • Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>_.:=+~@</code> • Cannot start with sys • Cannot start or end with a space.
value	No	String	Tag value. <ul style="list-style-type: none"> • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>_.:=+~@</code>

Response Parameters

None

Example Requests

Creating instance tags with tag keys key1 and key2 and tag values value1 and value2

```
POST https://{endpoint}/v2/{project_id}/kafka/{instance_id}/tags/action
{
  "action": "create",
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

Example Responses

None

SDK Sample Code

The SDK sample code is as follows.

Java

Creating instance tags with tag keys key1 and key2 and tag values value1 and value2

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateOrDeleteKafkaTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateOrDeleteKafkaTagRequest request = new BatchCreateOrDeleteKafkaTagRequest();
        request.withInstanceId("{instance_id}");
        BatchCreateOrDeleteTagReq body = new BatchCreateOrDeleteTagReq();
        List<TagEntity> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagEntity()
                .withKey("key1")
                .withValue("value1")
        );
        listbodyTags.add(
            new TagEntity()
                .withKey("key2")
                .withValue("value2")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchCreateOrDeleteTagReq.ActionEnum.fromValue("create"));
        request.withBody(body);
        try {
            BatchCreateOrDeleteKafkaTagResponse response = client.batchCreateOrDeleteKafkaTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Creating instance tags with tag keys key1 and key2 and tag values value1 and value2

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateOrDeleteKafkaTagRequest()
        request.instance_id = "{instance_id}"
        listTagsbody = [
            TagEntity(
                key="key1",
                value="value1"
            ),
            TagEntity(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchCreateOrDeleteTagReq(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_or_delete_kafka_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Creating instance tags with tag keys key1 and key2 and tag values value1 and value2

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := kafka.NewKafkaClient(  
        kafka.KafkaClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.BatchCreateOrDeleteKafkaTagRequest{}  
    request.InstanceId = "{instance_id}"  
    keyTags := "key1"  
    valueTags := "value1"  
    keyTags1 := "key2"  
    valueTags1 := "value2"  
    var listTagsbody = []model.TagEntity{  
        {  
            Key: &keyTags,  
            Value: &valueTags,  
        },  
        {  
            Key: &keyTags1,  
            Value: &valueTags1,  
        },  
    }  
    actionBatchCreateOrDeleteTagReq := model.GetBatchCreateOrDeleteTagReqActionEnum().CREATE  
    request.Body = &model.BatchCreateOrDeleteTagReq{  
        Tags: &listTagsbody,  
        Action: &actionBatchCreateOrDeleteTagReq,  
    }  
    response, err := client.BatchCreateOrDeleteKafkaTag(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
204	Tags are successfully added or deleted.

Error Codes

See [Error Codes](#).

5.10.2 Listing Tags of an Instance

Function

This API is used to query instance tags.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/{instance_id}/tags

Table 5-253 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-254 Response body parameters

Parameter	Type	Description
tags	Array of TagEntity objects	Tag list.

Table 5-255 TagEntity

Parameter	Type	Description
key	String	Tag key. <ul style="list-style-type: none"> • Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>_:=+@</code> • Cannot start with <i>sys</i> • Cannot start or end with a space.
value	String	Tag value. <ul style="list-style-type: none"> • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>_:=+@</code>

Example Requests

```
GET https://{endpoint}/v2/{project_id}/kafka/{instance_id}/tags
```

Example Responses

Status code: 200

The instance tags are listed successfully.

```
{
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```



```
public class ShowKafkaTagsSolution {
    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowKafkaTagsRequest request = new ShowKafkaTagsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowKafkaTagsResponse response = client.showKafkaTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
```

```

request = ShowKafkaTagsRequest()
request.instance_id = "{instance_id}"
response = client.show_kafka_tags(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaTagsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowKafkaTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The instance tags are listed successfully.

Error Codes

See [Error Codes](#).

5.10.3 Listing Tags of a Project

Function

This API is used to query project tags.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/tags

Table 5-256 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see Obtaining a Project ID .

Request Parameters

None

Response Parameters

Status code: 200

Table 5-257 Response body parameters

Parameter	Type	Description
tags	Array of TagMultyValueEntity objects	Tag list.

Table 5-258 TagMultyValueEntity

Parameter	Type	Description
key	String	Tag key.
values	Array of strings	Tag value.

Example Requests

```
GET https://{endpoint}/v2/{project_id}/kafka/tags
```

Example Responses

Status code: 200

The project tags are listed successfully.

```
{
  "tags": [ {
    "key": "key1",
    "values": [ "value-test", "value1" ]
  }, {
    "key": "key2",
    "values": [ "value2" ]
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowKafkaProjectTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowKafkaProjectTagsRequest request = new ShowKafkaProjectTagsRequest();
try {
    ShowKafkaProjectTagsResponse response = client.showKafkaProjectTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowKafkaProjectTagsRequest()
        response = client.show_kafka_project_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
```

```

)
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowKafkaProjectTagsRequest{}
    response, err := client.ShowKafkaProjectTags(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The project tags are listed successfully.

Error Codes

See [Error Codes](#).

5.11 Diagnosis Management

5.11.1 Pre-check Before Diagnosing Message Stack

Function

This API is used to pre-check before diagnosing message stack.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/diagnosis-check

Table 5-259 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 5-260 Query Parameters

Parameter	Mandatory	Type	Description
group	Yes	String	Consumer group name.
topic	Yes	String	Topic name.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-261 Response body parameters

Parameter	Type	Description
[items]	Array of KafkaDiagnosisCheckEntity objects	Returned object of Kafka message stack diagnosis pre-check.

Table 5-262 KafkaDiagnosisCheckEntity

Parameter	Type	Description
name	String	Pre-check item name.
reason	String	Pre-check failure cause.

Parameter	Type	Description
success	Boolean	Whether pre-check succeeded.

Status code: 400

Table 5-263 Response body parameters

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

Example Requests

Pre-checking topic-1 subscribed by consumer group group-1 before diagnosing message stack

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/diagnosis-check?group=group-1&topic=topic-1
```

Example Responses

Status code: 200

Successful

```
[ {
  "name" : "RAM",
  "success" : false,
  "reason" : "1,2"
}, {
  "name" : "CPU",
  "success" : false,
  "reason" : "1,2"
}, {
  "name" : "SUBSCRIPTION",
  "success" : false,
  "reason" : "0,2"
}, {
  "name" : "LAG",
  "success" : false,
  "reason" : "{\\"0\\":10159908,\\"1\\":9768464,\\"2\\":9361706}"
}, {
  "name" : "PRODUCE",
  "success" : false,
  "reason" : "{\\"1694571420000\\":14.159}"
} ]
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;
```



```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowDiagnosisPreCheckSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDiagnosisPreCheckRequest request = new ShowDiagnosisPreCheckRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowDiagnosisPreCheckResponse response = client.showDiagnosisPreCheck(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowDiagnosisPreCheckRequest()
    request.instance_id = "{instance_id}"
    response = client.show_diagnosis_pre_check(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDiagnosisPreCheckRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ShowDiagnosisPreCheck(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful
400	Invalid parameter

Error Codes

See [Error Codes](#).

5.11.2 Creating a Message Stack Diagnosis Task

Function

This API is used to create a message stack diagnosis task.

Calling Method

For details, see [Calling APIs](#).

URI

POST /v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks

Table 5-264 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-265 Request body parameters

Parameter	Mandatory	Type	Description
group_name	Yes	String	Consumer group name.
topic_name	Yes	String	Topic name.

Response Parameters

Status code: 200

Table 5-266 Response body parameters

Parameter	Type	Description
report_id	String	Diagnosis report ID.

Example Requests

Creating a message stack diagnosis task for consumer group group-test and topic topic-test

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks
{
  "group_name" : "group-test",
  "topic_name" : "topic-test"
}
```

Example Responses

Status code: 200

Message stack diagnosis task ID.

```
{
  "report_id" : "0e35a9f4-e75d-4fd6-b230-280860f666f7"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Creating a message stack diagnosis task for consumer group group-test and topic topic-test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateMessageDiagnosisTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
CreateMessageDiagnosisTaskRequest request = new CreateMessageDiagnosisTaskRequest();
request.withInstanceId("{instance_id}");
CreateMessageDiagnosisTaskReq body = new CreateMessageDiagnosisTaskReq();
body.withTopicName("topic-test");
body.withGroupName("group-test");
request.withBody(body);
try {
    CreateMessageDiagnosisTaskResponse response = client.createMessageDiagnosisTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Creating a message stack diagnosis task for consumer group group-test and topic topic-test

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateMessageDiagnosisTaskRequest()
        request.instance_id = "{instance_id}"
        request.body = CreateMessageDiagnosisTaskReq(
            topic_name="topic-test",
            group_name="group-test"
        )
```

```
response = client.create_message_diagnosis_task(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

Creating a message stack diagnosis task for consumer group group-test and topic topic-test

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateMessageDiagnosisTaskRequest{}
    request.InstanceId = "{instance_id}"
    request.Body = &model.CreateMessageDiagnosisTaskReq{
        TopicName: "topic-test",
        GroupName: "group-test",
    }
    response, err := client.CreateMessageDiagnosisTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Message stack diagnosis task ID.

Error Codes

See [Error Codes](#).

5.11.3 Querying the Message Stack Diagnosis Report List

Function

This API is used to query the message stack diagnosis report list.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks

Table 5-267 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 5-268 Query Parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset. The records after this offset will be queried.
limit	No	Integer	Maximum number of records that can be returned.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-269 Response body parameters

Parameter	Type	Description
report_list	Array of KafkaMessageDiagnosisReportInfoEntity objects	Diagnosis reports.
total_num	Integer	Total number of diagnosis reports.

Table 5-270 KafkaMessageDiagnosisReportInfoEntity

Parameter	Type	Description
report_id	String	Diagnosis report ID.
status	String	Status of a message stack diagnosis task. <ul style="list-style-type: none"> • diagnosing • failed • deleted (manually) • finished • normal: No problems found. • abnormal: Problems found.
begin_time	String	Diagnosis task start time.
end_time	String	Diagnosis task end time.
group_name	String	Name of the consumer group being diagnosed.
topic_name	String	Name of the topic being diagnosed.
accumulated_partitions	Integer	Number of partitions where accumulated messages are found.

Example Requests

Querying message stack diagnosis reports

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks?offset=1&limit=10
```

Example Responses

Status code: 200

Successful

```
{
  "report_list" : [ {
    "report_id" : "89b202d5-1f34-4a89-af9d-698496d0b7b1",
    "status" : "diagnosing",
    "begin_time" : "2022-11-11T10:00:00.000Z",
    "end_time" : "2022-11-11T10:00:00.000Z",
    "group_name" : "group-test",
    "topic_name" : "topic-test",
    "accumulated_partitions" : 3
  } ],
  "total_num" : 1
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListMessageDiagnosisReportsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListMessageDiagnosisReportsRequest request = new ListMessageDiagnosisReportsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListMessageDiagnosisReportsResponse response = client.listMessageDiagnosisReports(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListMessageDiagnosisReportsRequest()
        request.instance_id = "{instance_id}"
        response = client.list_message_diagnosis_reports(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
```

```

Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListMessageDiagnosisReportsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ListMessageDiagnosisReports(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.11.4 Batch Deleting Message Stack Diagnosis Reports

Function

This API is used to delete message stack diagnosis reports in batches.

Calling Method

For details, see [Calling APIs](#).

URI

DELETE /v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks

Table 5-271 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.

Request Parameters

Table 5-272 Request body parameters

Parameter	Mandatory	Type	Description
report_id_list	Yes	Array of strings	IDs of reports to be deleted.

Response Parameters

Status code: 200

Table 5-273 Response body parameters

Parameter	Type	Description
results	Array of results objects	Diagnosis report deletion report.

Table 5-274 results

Parameter	Type	Description
result	Boolean	Report deletion report.
id	String	Report ID.

Example Requests

Batch Deleting message stack diagnosis reports

```
DELETE https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis-tasks
{
  "report_id_list" : [ "0e35a9f4-e75d-4fd6-b230-280860f666f7" ]
}
```

Example Responses

Status code: 200

Successful

```
{
  "results" : [ {
```

```
"result" : "true",  
  "id" : "0e35a9f4-e75d-4fd6-b230-280860f666f7"  
}]  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

Batch Deleting message stack diagnosis reports

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class BatchDeleteMessageDiagnosisReportsSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
  
        BatchDeleteMessageDiagnosisReportsRequest request = new  
BatchDeleteMessageDiagnosisReportsRequest();  
        request.withInstanceId("{instance_id}");  
        BatchDeleteMessageDiagnosisReportsReq body = new BatchDeleteMessageDiagnosisReportsReq();  
        List<String> listbodyReportIdList = new ArrayList<>();  
        listbodyReportIdList.add("0e35a9f4-e75d-4fd6-b230-280860f666f7");  
        body.withReportIdList(listbodyReportIdList);  
        request.withBody(body);  
        try {  
            BatchDeleteMessageDiagnosisReportsResponse response =  
client.batchDeleteMessageDiagnosisReports(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
        }  
    }  
}
```

```
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

Batch Deleting message stack diagnosis reports

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchDeleteMessageDiagnosisReportsRequest()
        request.instance_id = "{instance_id}"
        listReportIdListbody = [
            "0e35a9f4-e75d-4fd6-b230-280860f666f7"
        ]
        request.body = BatchDeleteMessageDiagnosisReportsReq(
            report_id_list=listReportIdListbody
        )
        response = client.batch_delete_message_diagnosis_reports(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

Batch Deleting message stack diagnosis reports

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
// risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
// variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
// example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.BatchDeleteMessageDiagnosisReportsRequest{}
request.InstanceId = "{instance_id}"
var listReportIdListbody = []string{
    "0e35a9f4-e75d-4fd6-b230-280860f666f7",
}
request.Body = &model.BatchDeleteMessageDiagnosisReportsReq{
    ReportIdList: listReportIdListbody,
}
response, err := client.BatchDeleteMessageDiagnosisReports(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.11.5 Querying Diagnosis Report Details

Function

This API is used to query diagnosis report details.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis/{report_id}

Table 5-275 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
report_id	Yes	String	Message stack diagnosis report ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-276 Response body parameters

Parameter	Type	Description
abnormal_item_num	Integer	Total number of exceptions found in the diagnosis.
failed_item_num	Integer	Total number of failed diagnosis items.
normal_item_num	Integer	Total number of normal diagnosis items.
diagnosis_dimension_list	Array of KafkaMessageDiagnosisDimensionEntity objects	Diagnosis items.

Table 5-277 KafkaMessageDiagnosisDimensionEntity

Parameter	Type	Description
name	String	Diagnosis dimension name.

Parameter	Type	Description
abnormal_num	Integer	Total number of abnormal items in this diagnosis dimension.
failed_num	Integer	Total number of failed items in this diagnosis dimension.
diagnosis_item_list	Array of KafkaMessageDiagnosisItemEntity objects	Diagnosis dimensions.

Table 5-278 KafkaMessageDiagnosisItemEntity

Parameter	Type	Description
name	String	Diagnosis item name.
result	String	Diagnosis result.
cause_ids	Array of KafkaMessageDiagnosisConclusionEntity objects	Diagnosis exception causes.
advice_ids	Array of KafkaMessageDiagnosisConclusionEntity objects	Suggestions for diagnosis exceptions.
partitions	Array of integers	Partitions affected by the diagnosis exceptions.
failed_partitions	Array of integers	Partitions that failed to be diagnosed.
broker_ids	Array of integers	Brokers affected by the diagnosis exceptions.

Table 5-279 KafkaMessageDiagnosisConclusionEntity

Parameter	Type	Description
id	Integer	Diagnosis conclusion ID.
params	Map<String,String>	Diagnosis conclusion parameters.

Example Requests

Querying diagnosis report details

```
GET https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/message-diagnosis/{report_id}
```

Example Responses

Status code: 200

Successful

```
{
  "abnormal_item_num" : 2,
  "failed_item_num" : 2,
  "normal_item_num" : 1,
  "diagnosis_dimension_list" : [ {
    "name" : "server",
    "abnormal_num" : 1,
    "failed_num" : 1,
    "diagnosis_item_list" : [ {
      "name" : "fetch_error",
      "result" : "abnormal",
      "cause_ids" : [ {
        "id" : 1,
        "params" : {
          "THRESHOLD" : "90",
          "ERROR_CODE" : "[1,2,3]"
        }
      }
    ]
  } ],
  "advice_ids" : [ {
    "id" : 1,
    "params" : { }
  } ],
  "partitions" : [ 1, 2, 3 ],
  "failed_partitions" : [ 4, 5, 6 ],
  "broker_ids" : [ 1, 2, 3 ]
} ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowMessageDiagnosisReportSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";
```

```
ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowMessageDiagnosisReportRequest request = new ShowMessageDiagnosisReportRequest();
request.withInstanceId("{instance_id}");
request.withReportId("{report_id}");
try {
    ShowMessageDiagnosisReportResponse response = client.showMessageDiagnosisReport(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMessageDiagnosisReportRequest()
        request.instance_id = "{instance_id}"
        request.report_id = "{report_id}"
        response = client.show_message_diagnosis_report(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMessageDiagnosisReportRequest{}
    request.InstanceId = "{instance_id}"
    request.ReportId = "{report_id}"
    response, err := client.ShowMessageDiagnosisReport(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

5.12 Other APIs

5.12.1 Listing Maintenance Time Windows

Function

This API is used to query the start time and end time of maintenance time windows.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/instances/maintain-windows

Request Parameters

None

Response Parameters

Status code: 200

Table 5-280 Response body parameters

Parameter	Type	Description
maintain_windows	Array of MaintainWindowsEntity objects	List of supported maintenance time windows.

Table 5-281 MaintainWindowsEntity

Parameter	Type	Description
default	Boolean	Whether the maintenance time window is set to the default time segment.
end	String	End time of the maintenance time window.
begin	String	Start time of the maintenance time window.
seq	Integer	Sequence number.

Example Requests

```
GET https://{endpoint}/v2/instances/maintain-windows
```

Example Responses

Status code: 200

Query succeeded.

```
{
  "maintain_windows" : [ {
    "default" : false,
    "seq" : 1,
    "begin" : "22",
    "end" : "02"
  }, {
    "default" : true,
    "seq" : 2,
    "begin" : "02",
    "end" : "06"
  }, {
    "default" : false,
    "seq" : 3,
    "begin" : "06",
    "end" : "10"
  }, {
    "default" : false,
    "seq" : 4,
    "begin" : "10",
    "end" : "14"
  }, {
    "default" : false,
    "seq" : 5,
    "begin" : "14",
    "end" : "18"
  }, {
    "default" : false,
    "seq" : 6,
    "begin" : "18",
    "end" : "22"
  }
  ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowMaintainWindowsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
```

```
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ShowMaintainWindowsRequest request = new ShowMaintainWindowsRequest();
try {
    ShowMaintainWindowsResponse response = client.showMaintainWindows(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowMaintainWindowsRequest()
        response = client.show_maintain_windows(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowMaintainWindowsRequest{}
    response, err := client.ShowMaintainWindows(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Query succeeded.

Error Codes

See [Error Codes](#).

5.12.2 Listing AZ Information

Function

This API is used to query the AZ ID for creating an instance.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/available-zones

Request Parameters

None

Response Parameters

Status code: 200

Table 5-282 Response body parameters

Parameter	Type	Description
region_id	String	Region ID.
available_zones	Array of AvailableZonesResp objects	Array of AZs.

Table 5-283 AvailableZonesResp

Parameter	Type	Description
soldOut	Boolean	Indicates whether resources are sold out.
id	String	AZ ID.
code	String	AZ code.
name	String	AZ name.
port	String	AZ port.
resource_availability	String	Indicates whether the AZ has available resources.
default_az	Boolean	Indicates whether the AZ is the default AZ.

Parameter	Type	Description
remain_time	Long	Remaining time.
ipv6_enable	Boolean	Indicates whether IPv6 is supported.

Example Requests

```
GET https://{endpoint}/v2/available-zones
```

Example Responses

Status code: 200

The AZ information is queried successfully.

```
{
  "region_id": "xxx",
  "available_zones": [ {
    "soldOut": false,
    "id": "d539378ec1314c85b76fefa3f7071458",
    "code": "xxx",
    "name": "AZ 2.",
    "port": "8003",
    "resource_availability": "true",
    "default_az": true,
    "remain_time": 9223372036854776000,
    "ipv6_enable": false
  }, {
    "soldOut": false,
    "id": "9f1c5806706d4c1fb0eb72f0a9b18c77",
    "code": "xxx",
    "name": "AZ 3.",
    "port": "443",
    "resource_availability": "true",
    "default_az": false,
    "remain_time": 9223372036854776000,
    "ipv6_enable": false
  } ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ListAvailableZonesSolution {
    public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
ListAvailableZonesRequest request = new ListAvailableZonesRequest();
try {
    ListAvailableZonesResponse response = client.listAvailableZones(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAvailableZonesRequest()
        response = client.list_available_zones(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAvailableZonesRequest{}
    response, err := client.ListAvailableZones(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The AZ information is queried successfully.

Error Codes

See [Error Codes](#).

5.12.3 Querying Product Specifications List

Function

This API is used to query the product specifications list.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{engine}/products

Table 5-284 Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine.

Table 5-285 Query Parameters

Parameter	Mandatory	Type	Description
product_id	No	String	Product ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-286 Response body parameters

Parameter	Type	Description
engine	String	Message engine of DMS.
versions	Array of strings	Supported versions.
products	Array of ListEngineProductsEntity objects	Product specification details.

Table 5-287 ListEngineProductsEntity

Parameter	Type	Description
type	String	Product type. Currently, single-node and cluster types are supported.
product_id	String	Product ID.
ecs_flavor_id	String	ECS flavor.
billing_code	String	Billing mode.
arch_types	Array of strings	CPU architecture.
charging_mode	Array of strings	Billing mode. monthly : yearly/monthly; hourly : pay-per-use
ios	Array of ListEngineIosEntity objects	List of supported disk I/O types.
support_features	Array of ListEngineSupportFeaturesEntity objects	List of features supported by instances of the current specifications.
properties	ListEnginePropertiesEntity object	Attribute of instances of the current specifications.

Table 5-288 ListEngineIosEntity

Parameter	Type	Description
io_spec	String	Disk I/O code.
type	String	Disk type.
available_zones	Array of strings	Available AZs.
unavailable_zones	Array of strings	Unavailable AZs.

Table 5-289 ListEngineSupportFeaturesEntity

Parameter	Type	Description
name	String	Feature name.
properties	ListEngineSupportFeaturesPropertiesEntity object	Description of the features supported by the instance.

Table 5-290 ListEngineSupportFeaturesPropertiesEntity

Parameter	Type	Description
max_task	String	Maximum number of dumping tasks.
min_task	String	Minimum number of dumping tasks.
max_node	String	Maximum number of dumping nodes.
min_node	String	Minimum number of dumping nodes.

Table 5-291 ListEnginePropertiesEntity

Parameter	Type	Description
max_partition_per_broker	String	Maximum number of partitions of each broker.
max_broker	String	Maximum number of brokers.
max_storage_per_node	String	Maximum storage space of each broker. The unit is GB.
max_consumer_per_broker	String	Maximum number of consumers of each broker.
min_broker	String	Minimum number of brokers.
max_bandwidth_per_broker	String	Maximum bandwidth of each broker.
min_storage_per_node	String	Minimum storage space of each broker. The unit is GB.
max_tps_per_broker	String	Maximum TPS of each broker.
product_alias	String	Alias of product_id .

Example Requests

```
GET https://{endpoint}/v2/kafka/products
```

Example Responses

Status code: 200

The product specifications are listed successfully.

```
{
  "engine": "kafka",
  "versions": [ "1.1.0", "2.3.0" ],
  "products": [ {
    "type": "cluster",
    "product_id": "c6.2u4g.cluster",
    "ecs_flavor_id": "c6.large.2",

```

```
"billing_code" : "dms.platinum.c6",
"arch_types" : [ "X86" ],
"charging_mode" : [ "monthly", "hourly" ],
"ios" : [ {
  "io_spec" : "dms.physical.storage.high.v2",
  "type" : "evs",
  "available_zones" : [ "xxx", "xxx" ],
  "unavailable_zones" : [ "xxx", "xxx" ]
}, {
  "io_spec" : "dms.physical.storage.ultra.v2",
  "type" : "evs",
  "available_zones" : [ "xxx", "xxx" ],
  "unavailable_zones" : [ "xxx", "xxx" ]
}],
"support_features" : [ {
  "name" : "connector_obs",
  "properties" : {
    "max_task" : "10",
    "max_node" : "10",
    "min_task" : "1",
    "min_node" : "2"
  }
} ],
"properties" : {
  "max_partition_per_broker" : "250",
  "max_broker" : "30",
  "max_storage_per_node" : "10000",
  "max_consumer_per_broker" : "4000",
  "min_broker" : "3",
  "max_bandwidth_per_broker" : "100",
  "min_storage_per_node" : "200",
  "max_tps_per_broker" : "30000",
  "product_alias" : "kafka.2u4g.cluster"
} ],
{
  "type" : "cluster",
  "product_id" : "c6.4u8g.cluster",
  "ecs_flavor_id" : "c6.xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "500",
    "max_broker" : "30",
    "max_storage_per_node" : "20000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "400",
```



```
"max_tps_per_broker" : "100000",
"product_alias" : "kafka.4u8g.cluster"
}
}, {
  "type" : "cluster",
  "product_id" : "c6.8u16g.cluster",
  "ecs_flavor_id" : "c6.2xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "1000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "800",
    "max_tps_per_broker" : "150000",
    "product_alias" : "kafka.8u16g.cluster"
  }
}, {
  "type" : "cluster",
  "product_id" : "c6.12u24g.cluster",
  "ecs_flavor_id" : "c6.3xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
```

```
"max_partition_per_broker" : "1500",
"max_broker" : "30",
"max_storage_per_node" : "30000",
"max_consumer_per_broker" : "4000",
"min_broker" : "3",
"max_bandwidth_per_broker" : "100",
"min_storage_per_node" : "1200",
"max_tps_per_broker" : "200000",
"product_alias" : "kafka.12u24g.cluster"
}
}, {
  "type" : "cluster",
  "product_id" : "c6.16u32g.cluster",
  "ecs_flavor_id" : "c6.4xlarge.2",
  "billing_code" : "dms.platinum.c6",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "type" : "evs",
    "available_zones" : [ "xxx", "xxx" ],
    "unavailable_zones" : [ "xxx", "xxx" ]
  } ],
  "support_features" : [ {
    "name" : "connector_obs",
    "properties" : {
      "max_task" : "10",
      "max_node" : "10",
      "min_task" : "1",
      "min_node" : "2"
    }
  } ],
  "properties" : {
    "max_partition_per_broker" : "2000",
    "max_broker" : "30",
    "max_storage_per_node" : "30000",
    "max_consumer_per_broker" : "4000",
    "min_broker" : "3",
    "max_bandwidth_per_broker" : "100",
    "min_storage_per_node" : "1600",
    "max_tps_per_broker" : "250000",
    "product_alias" : "kafka.16u32g.cluster"
  }
}
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;
```

```
public class ListEngineProductsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEngineProductsRequest request = new ListEngineProductsRequest();
        request.withEngine(ListEngineProductsRequest.EngineEnum.fromValue("{engine}"));
        try {
            ListEngineProductsResponse response = client.listEngineProducts(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListEngineProductsRequest()
        request.engine = "{engine}"
        response = client.list_engine_products(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEngineProductsRequest{}
    request.Engine = model.GetListEngineProductsRequestEngineEnum().ENGINE
    response, err := client.ListEngineProducts(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	The product specifications are listed successfully.

Error Codes

See [Error Codes](#).

5.12.4 Querying Kafka Instance Monitoring Dimensions

Function

This API is used to query Kafka instance monitoring dimensions.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/{project_id}/instances/{instance_id}/ces-hierarchy

Table 5-292 Path Parameters

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

Request Parameters

None

Response Parameters

Status code: 200

Table 5-293 Response body parameters

Parameter	Type	Description
dimensions	Array of dimensions objects	Monitoring dimensions.
instance_ids	Array of instance_ids objects	Instance information.
nodes	Array of nodes objects	Broker information.

Parameter	Type	Description
queues	Array of queues objects	Topic information.
groups	Array of groups objects	Consumer group information.

Table 5-294 dimensions

Parameter	Type	Description
name	String	Monitoring dimension name.
metrics	Array of strings	Metric name.
key_name	Array of strings	Key used for monitoring query.
dim_router	Array of strings	Monitoring dimension route.
children	Array of children objects	List of secondary dimensions.

Table 5-295 children

Parameter	Type	Description
name	String	Secondary dimension name.
metrics	Array of strings	Metrics on the secondary dimension.
key_name	Array of strings	Key used for monitoring query.
dim_router	Array of strings	Monitoring dimension route.

Table 5-296 instance_ids

Parameter	Type	Description
name	String	Instance ID.

Table 5-297 nodes

Parameter	Type	Description
name	String	Broker name.

Table 5-298 queues

Parameter	Type	Description
name	String	Topic name.
partitions	Array of partitions objects	Partition list.

Table 5-299 partitions

Parameter	Type	Description
name	String	Partition name.

Table 5-300 groups

Parameter	Type	Description
name	String	Consumer group name.
queues	Array of queues objects	Topic information.

Table 5-301 queues

Parameter	Type	Description
name	String	Topic name.
partitions	Array of partitions objects	Partition information.

Table 5-302 partitions

Parameter	Type	Description
name	String	Partition name.

Example Requests

GET https://{endpoint}/v2/{project_id}/instances/{instance_id}/ces-hierarchy

Example Responses

Status code: 200

Query succeeded.

```

{
  "dimensions": [ {
    "name": "kafka_instance_id",
    "metrics": [ "current_partitions", "current_topics", "group_messages" ],
    "key_name": [ "instance_ids" ],
    "dim_router": [ "kafka_instance_id" ]
  }, {
    "name": "kafka_broker",
    "metrics": [ "broker_data_size", "broker_messages_in_rate", "broker_bytes_out_rate",
"broker_bytes_in_rate", "broker_produce_mean", "broker_fetch_mean" ],
    "key_name": [ "nodes" ],
    "dim_router": [ "kafka_instance_id", "kafka_broker" ]
  }, {
    "name": "kafka_rest",
    "metrics": [ "rest_produce_success", "rest_produce_failed", "rest_produce_latency",
"rest_produce_msg_num", "rest_produce_flow", "rest_consume_success", "rest_consume_failed",
"rest_consume_latency", "rest_consume_msg_num", "rest_consume_flow", "rest_commit_success",
"rest_commit_failed", "rest_commit_latency", "rest_commit_msg_num", "rest_commit_flow" ],
    "key_name": [ "nodes" ],
    "dim_router": [ "kafka_instance_id", "kafka_rest" ]
  }, {
    "name": "kafka_topics",
    "metrics": [ "topic_data_size", "topic_messages_in_rate", "topic_bytes_out_rate", "topic_bytes_in_rate",
"topic_messages" ],
    "key_name": [ "queues" ],
    "dim_router": [ "kafka_instance_id", "kafka_topics" ],
    "children": [ {
      "name": "kafka_partitions",
      "metrics": [ "produced_messages", "partition_messages" ],
      "key_name": [ "queues", "partitions" ],
      "dim_router": [ "kafka_instance_id", "kafka_topics", "kafka_partitions" ]
    } ]
  }, {
    "name": "kafka_groups_partitions",
    "metrics": [ "messages_consumed", "messages_remained" ],
    "key_name": [ "groups", "queues", "partitions" ],
    "dim_router": [ "kafka_instance_id", "kafka_groups", "kafka_groups_topics", "kafka_groups_partitions" ]
  } ],
  "instance_ids": [ {
    "name": "68f3f6a0-3741-453b-bda9-a6ff6b5bb6f7"
  } ],
  "nodes": [ {
    "name": "broker-0"
  }, {
    "name": "broker-1"
  }, {
    "name": "broker-2"
  } ],
  "queues": [ {
    "name": "aaaa",
    "partitions": [ {
      "name": "0"
    } ]
  }, {
    "name": "mytest",
    "partitions": [ {
      "name": "0"
    }, {
      "name": "1"
    }, {
      "name": "2"
    } ]
  }, {
    "name": "topic-84234378",
    "partitions": [ {
      "name": "0"
    }, {
      "name": "1"
    } ],
  } ],

```



```
    "name" : "2"
  } ]
} ],
"groups" : [ {
  "name" : "test-consumer-group",
  "queues" : [ {
    "name" : "mytest",
    "partitions" : [ {
      "name" : "0"
    }, {
      "name" : "1"
    }, {
      "name" : "2"
    } ]
  } ]
} ]
} ]
} ]
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowCesHierarchySolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowCesHierarchyRequest request = new ShowCesHierarchyRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowCesHierarchyResponse response = client.showCesHierarchy(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowCesHierarchyRequest()
        request.instance_id = "{instance_id}"
        response = client.show_ces_hierarchy(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
```

```

WithAk(ak).
WithSk(sk).
WithProjectId(projectId).
Build()

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowCesHierarchyRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowCesHierarchy(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

More

For SDK sample code of more programming languages, see the Sample Code tab in [API Explorer](#). SDK sample code can be automatically generated.

Status Codes

Status Code	Description
200	Query succeeded.

Error Codes

See [Error Codes](#).

5.12.5 Querying vCPUs of a Kafka Flavor

Function

This API is used to query the number of vCPUs of a Kafka flavor.

Calling Method

For details, see [Calling APIs](#).

URI

GET /v2/kafka/products/cores

Table 5-303 Query Parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.
product_id	Yes	String	Product ID.

Request Parameters

None

Response Parameters

Status code: 200

Table 5-304 Response body parameters

Parameter	Type	Description
core_num	Integer	Number of vCPUs.

Example Requests

```
GET https://{endpoint}/v2/kafka/products/cores
```

Example Responses

Status code: 200

Successful

```
{  
  "core_num" : 100  
}
```

Status Codes

Status Code	Description
200	Successful

Error Codes

See [Error Codes](#).

6 Permissions and Supported Actions

This chapter describes fine-grained permissions management for your Kafka instances. If your Huawei ID does not need individual IAM users, then you may skip over this chapter.

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

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

For details about DMS for Kafka system policies, see [Permissions Management](#).

NOTE

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

An account has all of the permissions required to call all APIs, but IAM users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query Kafka instances using an API, the user must have been granted permissions that allow the **dms:instance:create** action.

Supported Actions

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

- **Permission:** a statement in a policy that allows or denies certain operations.
- **APIs:** REST APIs that can be called by a user who has been granted specific permissions.

- Action: Specific operations that are allowed or denied.
- IAM projects or enterprise projects: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions for both IAM and enterprise projects can be used and take effect for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM. Administrators can check whether an action supports IAM projects or enterprise projects in the action list. For details about the differences between IAM and enterprise projects, see [What Are the Differences Between IAM and Enterprise Management?](#)

DMS for Kafka supports the following actions in custom policies:

- **Lifecycle management actions**, including actions supported by Kafka instance lifecycle management APIs, such as the APIs for creating an instance, querying the instance list, modifying instance information, and batch restarting or deleting instances.
- **Instance management actions**, including actions supported by Kafka instance management APIs, such as the APIs for resetting passwords and configuring automatic topic creation.
- **Smart Connect actions**, including actions supported by Smart Connect APIs, such as the APIs for enabling or disabling Smart Connect, creating a Smart Connect task.
- **Specification modification management action**, supported by the specification modification management API for modifying instance specifications.
- **Topic management actions**, including actions supported by topic management APIs, such as the APIs for creating, querying, and modifying topics.
- **User management actions**, including actions supported by user management APIs, such as the APIs for creating users, querying users, and configuring user permissions.
- **Message query actions**, including actions supported by message query APIs, such as the API for querying messages.
- **Background task management actions**, including actions supported by background task management APIs, such as the APIs for querying the background task list of an instance and querying a specified background task.
- **Tag management actions**, including actions supported by tag management APIs, such as the APIs for querying instance tags and project tags.

Lifecycle Management

Table 6-1 Lifecycle management

Permission	API	Action	IAM (Project)	Enterprise (Enterprise Project)
Creating an instance	POST /v2/{engine}/{project_id}/instances	dms:instance:create	√	√
Querying all instances	GET /v2/{project_id}/instances	dms:instance:list	√	√
Querying an instance	GET /v2/{project_id}/instances/{instance_id}	dms:instance:get	√	√
Deleting an instance	DELETE /v2/{project_id}/instances/{instance_id}	dms:instance:delete	√	√
Modifying instance information	PUT /v2/{project_id}/instances/{instance_id}	dms:instance:modify	√	√
Batch restarting or deleting instances	POST /v2/{project_id}/instances/action	Restart: dms:instance:modifyStatus Delete: dms:instance:delete	√	√

Instance Management

Table 6-2 Instance management

Permission	API	Action	IAM (Project)	Enterprise (Enterprise Project)
Resetting a password	POST /v2/{project_id}/instances/{instance_id}/password	dms:instance:resetAuthInfo	√	√

Permission	API	Action	IAM (Project)	Enterprise (Enterprise Project)
Resetting the Kafka Manager password	PUT /v2/{project_id}/instances/{instance_id}/kafka-manager-password	dms:instance:resetAuthInfo	√	√
Restarting Kafka Manager	PUT /v2/{project_id}/instances/{instance_id}/restart-kafka-manager	dms:instance:modifyStatus	√	√
Configuring automatic topic creation	POST /v2/{project_id}/instances/{instance_id}/autotopic	dms:instance:modify	√	√
Modifying the private IP address for cross-VPC access	POST /v2/{project_id}/instances/{instance_id}/crossvpc/modify	dms:instance:modify	√	√
Resetting consumer group offset to the specified position	POST /v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset	dms:instance:modify	√	√

Smart Connect

Table 6-3 Smart Connect

Permission	API	Action	IAM Projects	Enterprise Projects
Enabling Smart Connect	POST /v2/{project_id}/instances/{instance_id}/connector	dms:instance:connector	√	√

Permission	API	Action	IAM Projects	Enterprise Projects
Disabling Smart Connect	POST /v2/{project_id}/kafka/instances/{instance_id}/delete-connector	dms:instance:connector	√	√
Creating a Smart Connect task	POST /v2/{project_id}/instances/{instance_id}/connector/tasks	dms:instance:createConnectorSinkTask	√	√
Listing Smart Connect tasks	GET /v2/{project_id}/instances/{instance_id}/connector/tasks	dms:instance:listConnectorSinkTask	√	√
Querying Smart Connect task details	GET /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}	dms:instance:getConnectorSinkTask	√	√
Deleting Smart Connect tasks	DELETE /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}	dms:instance:deleteConnectorSinkTask	√	√
Pausing Smart Connect tasks	PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/pause	dms:instance:updateConnectorTask	√	√
Restarting Smart Connect tasks	PUT /v2/{project_id}/instances/{instance_id}/connector/tasks/{task_id}/resume	dms:instance:updateConnectorTask	√	√
Starting a Smart Connect task or restarting a paused or running Smart Connect task	PUT /v2/{project_id}/kafka/instances/{instance_id}/connector/tasks/{task_id}/restart	dms:instance:updateConnectorTask	√	√

Specification Modification Management

Table 6-4 Specification modification management

Permission	API	Action	IAM Projects	Enterprise Projects
Modifying instance specifications	POST /v2/{engine}/{project_id}/instances/{instance_id}/extends	dms:instance:scale	√	√

Topic Management

Table 6-5 Topic management

Permission	API	Action	IAM Projects	Enterprise Projects
Creating a topic in a Kafka instance	POST /v2/{project_id}/instances/{instance_id}/topics	dms:instance:modify	√	√
Querying a topic in a Kafka instance	GET /v2/{project_id}/instances/{instance_id}/topics	dms:instance:get	√	√
Modifying topics of a Kafka instance	PUT /v2/{project_id}/instances/{instance_id}/topics	dms:instance:modify	√	√
Deleting topics in a Kafka instance in batches	POST /v2/{project_id}/instances/{instance_id}/topics/delete	dms:instance:modify	√	√

User Management

Table 6-6 User management

Permission	API	Action	IAM Projects	Enterprise Projects
Querying the user list	GET /v2/{project_id}/instances/{instance_id}/users	dms:instance:get	√	√
Creating a user	POST /v2/{project_id}/instances/{instance_id}/users	dms:instance:modify	√	√
Deleting users in batches	PUT /v2/{project_id}/instances/{instance_id}/users	dms:instance:modify	√	√
Resetting a user password	PUT /v2/{project_id}/instances/{instance_id}/users/{user_name}	dms:instance:get	√	√
Querying user permissions	GET /v1/{project_id}/instances/{instance_id}/topics/{topic_name}/accesspolicy	dms:instance:get	√	√
Granting user permissions	POST /v1/{project_id}/instances/{instance_id}/topics/accesspolicy	dms:instance:modify	√	√

Message Query

Table 6-7 Message query

Permission	API	Action	IAM Projects	Enterprise Projects
Querying messages	GET /v2/{project_id}/instances/{instance_id}/messages	dms:instance:get	√	√

Background Task Management

Table 6-8 Background task management

Permission	API	Action	IAM Projects	Enterprise Projects
Listing background tasks	GET /v2/{project_id}/instances/{instance_id}/tasks	dms:instance:getBackgroundTask	√	√
Querying a background task	GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	dms:instance:getBackgroundTask	√	√
Deleting a background task	DELETE /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	dms:instance:deleteBackgroundTask	√	√

Tag Management

Table 6-9 Tag management

Permission	API	Action	IAM Projects	Enterprise Projects
Batch adding or deleting tags	POST /v2/{project_id}/kafka/{instance_id}/tags/action	dms:instance:modify	√	√
Listing tags of an instance	GET /v2/{project_id}/kafka/{instance_id}/tags	dms:instance:get	√	√
Listing tags of a project	GET /v2/{project_id}/kafka/tags	dms:instance:get	√	√

7 Out-of-Date APIs

7.1 API V1

7.1.1 APIs for Managing Instances

7.1.1.1 Creating an Instance

 NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Creating an Instance](#).

Function

This API is used to create a **pay-per-use** instance.

URI

POST /v1.0/{*project_id*}/instances

[Table 7-1](#) describes the parameter.

Table 7-1 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

Request

Request parameters

[Table 7-2](#) describes the parameters.

Table 7-2 Request parameters

Parameter	Type	Mandatory	Description
name	String	Yes	Indicates the instance name. An instance name starts with a letter, consists of 4 to 64 characters, and can contain only letters, digits, underscores (_), and hyphens (-).
description	String	No	Indicates the description of an instance. It is a character string containing not more than 1024 characters. NOTE The backslash (\) and quotation mark (") are special characters for JSON packets. When using these characters in a parameter value, add the escape character (\) before these characters, for example, \\ and \".
engine	String	Yes	Indicates the message engine. Set the value to kafka .
engine_version	String	Yes	Indicates the version of the message engine.
specification	String	Yes	Indicates the baseline bandwidth of a Kafka instance, that is, the maximum amount of data transferred per unit time. Unit: MB Options: <ul style="list-style-type: none"> • 100 MB • 300 MB • 600 MB • 1200 MB
storage_space	Integer	Yes	Indicates the message storage space. Unit: GB. Value range: <ul style="list-style-type: none"> • Kafka instance with specification being 100MB: 600–90,000 GB • Kafka instance with specification being 300MB: 1200–90,000 GB • Kafka instance with specification being 600MB: 2400–90,000 GB • Kafka instance with specification being 1200MB: 4800–90,000 GB

Parameter	Type	Mandatory	Description
partition_num	Integer	Yes	<p>Indicates the maximum number of partitions in a Kafka instance.</p> <p>Options:</p> <ul style="list-style-type: none"> • When specification is 100MB: 300 • When specification is 300MB: 900 • When specification is 600MB: 1800 • When specification is 1200MB: 1800
access_user	String	No	<p>This parameter is mandatory when ssl_enable is set to true. This parameter is invalid when ssl_enable is set to false.</p> <p>Indicates a username. A username consists of 4 to 64 characters and can contain letters, digits, and hyphens (-).</p>
password	String	No	<p>This parameter is mandatory when ssl_enable is set to true. This parameter is invalid when ssl_enable is set to false.</p> <p>Indicates an instance password.</p> <p>The password must meet the following complexity requirements:</p> <ul style="list-style-type: none"> • Must be a string consisting of 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters `~!@#\$%^&*()-_+=+ [{}];',<.>/?
vpc_id	String	Yes	<p>Indicates the VPC ID.</p> <p>Obtain the value by using either of the following methods:</p> <ul style="list-style-type: none"> • Method 1: Log in to VPC console and view the VPC ID in the VPC details. • Method 2: Query the VPC ID through the VPC API. For details, see Querying VPCs.

Parameter	Type	Mandatory	Description
security_group_id	String	Yes	<p>Indicates the security group which the instance belongs to.</p> <p>Obtain the value by using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to VPC console. Choose Access Control > Security Groups in the navigation pane on the left. On the displayed page, click the target security group. You can view the security group ID on the displayed page. Method 2: Call the API for querying security groups. For details, see Querying Security Groups.
subnet_id	String	Yes	<p>Indicates the subnet ID.</p> <p>Obtain the value by using either of the following methods:</p> <ul style="list-style-type: none"> Method 1: Log in to VPC console and click the target subnet on the Subnets tab page. You can view the network ID on the displayed page. Method 2: Call the API for querying subnets. For details, see Querying Subnets.
available_zones	Array	Yes	<p>Indicates the ID of the AZ where brokers reside and which has available resources. The parameter value cannot be an empty array or an empty array. For details on how to obtain the value, see Querying AZ Information. Check whether the AZ has available resources.</p> <p>When creating a Kafka instance, you can select either 1 AZ or at least 3 AZ. When specifying AZs for brokers, use commas (,) to separate multiple AZs. Example parameter settings:</p> <ul style="list-style-type: none"> One AZ: "available_zones": ["a0865121f83b41cbafce65930a22a6e8"] Three or more AZs: "available_zones": ["a0865121f83b41cbafce65930a22a6e8","a0865121f83b41cbafce65930a22a6e7","a0865121f83b41cbafce65930a22a6e6"]
product_id	String	Yes	<p>Indicates the product ID.</p> <p>For details on how to obtain the ID, see Querying Product Specifications.</p>

Parameter	Type	Mandatory	Description
kafka_manager_user	String	Yes	Indicates the username for logging in to Kafka Manager. The username consists of 4 to 64 characters and can contain letters, digits, hyphens (-), and underscores (_).
kafka_manager_password	String	Yes	Indicates the password for logging in to Kafka Manager. The password must meet the following complexity requirements: <ul style="list-style-type: none"> • Must be a string consisting of 8 to 32 characters. • Must contain at least three of the following character types: <ul style="list-style-type: none"> - Lowercase letters - Uppercase letters - Digits - Special characters `~!@#%&*()-_=+\ [{}];',<.>/?
maintain_begin	String	No	Indicates the time at which a maintenance time window starts. Format: HH:mm:ss <ul style="list-style-type: none"> • The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window. For details about how to query the time segments of supported maintenance time windows, see Querying Maintenance Time Windows. • The start time must be set to 22:00:00, 02:00:00, 06:00:00, 10:00:00, 14:00:00, or 18:00:00. • Parameters maintain_begin and maintain_end must be set in pairs. If parameter maintain_begin is left blank, parameter maintain_end is also left blank. In this case, the system automatically sets the start time to 02:00:00.

Parameter	Type	Mandatory	Description
maintain_end	String	No	<p>Indicates the time at which a maintenance time window ends.</p> <p>Format: HH:mm:ss</p> <ul style="list-style-type: none"> The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window. For details about how to query the time segments of supported maintenance time windows, see Querying Maintenance Time Windows. The end time is four hours later than the start time. For example, if the start time is 22:00:00, the end time is 02:00:00. Parameters maintain_begin and maintain_end must be set in pairs. If parameter maintain_end is left blank, parameter maintain_start is also blank. In this case, the system automatically sets the end time to 06:00:00.
enable_publicip	Boolean	No	<p>Indicates whether to enable public access for an instance.</p> <ul style="list-style-type: none"> true: enable false: disable
public_bandwidth	String	No	<p>Indicates the public network bandwidth. Unit: Mbit/s</p> <p>Value range:</p> <ul style="list-style-type: none"> When specification is 100MB, the value must be a multiple of the number of brokers and fall in the range from 3 to 900. When specification is 300MB, the value must be a multiple of the number of brokers and fall in the range from 3 to 900. When specification is 600MB, the value must be a multiple of the number of brokers and fall in the range from 4 to 1200. When specification is 1200MB, the value must be a multiple of the number of brokers and fall in the range from 8 to 2400.

Parameter	Type	Mandatory	Description
publicip_id	String	No	Indicates the ID of the elastic IP address (EIP) bound to an instance. Use commas (,) to separate multiple EIP IDs. This parameter is mandatory if public access is enabled (that is, enable_publicip is set to true).
ssl_enable	Boolean	No	Indicates whether to enable SSL-encrypted access. <ul style="list-style-type: none"> • true: enable • false: disable
retention_policy	String	No	Indicates the action to be taken when the memory usage reaches the disk capacity threshold. Options: <ul style="list-style-type: none"> • time_base: Automatically delete the earliest messages. • produce_reject: Stop producing new messages.
enable_auto_topic	Boolean	No	Indicates whether to enable automatic topic creation. <ul style="list-style-type: none"> • true: enable • false: disable <p>If automatic topic creation is enabled, a topic will be automatically created with 3 partitions and 3 replicas when a message is produced to or consumed from a topic that does not exist.</p>
storage_spec_code	String	Yes	Indicates storage I/O specification. For details on how to select a disk type, see Disk Types and Performance . Options: <ul style="list-style-type: none"> • dms.physical.storage.high or dms.physical.storage.ultra when the parameter specification is 100MB • dms.physical.storage.high or dms.physical.storage.ultra when the parameter specification is 300MB • dms.physical.storage.ultra when the parameter specification is 600MB • dms.physical.storage.ultra when the parameter specification is 1200MB
enterprise_project_id	String	No	Indicates the enterprise project ID.

Parameter	Type	Mandatory	Description
tags	Array<Object>	No	Indicates the list of tags.

Table 7-3 tags

Parameter	Type	Mandatory	Description
key	String	No	Indicates the tag key. A tag key can contain a maximum of 36 Unicode characters. The key cannot be left blank or be an empty string. It cannot contain nonprintable ASCII (0-31) characters and the following special characters: =*<>\\, /
value	String	No	Indicates the value. A tag value can contain a maximum of 43 Unicode characters. The value cannot be left blank or be an empty string. It cannot contain nonprintable ASCII (0-31) characters and the following special characters: =*<>\\, /

```
{
  "name": "kafka-test",
  "engine": "kafka",
  "engine_version": "2.3.0",
  "specification": "100MB",
  "storage_space": 600,
  "partition_num": 300,
  "vpc_id": "b50c1aa7-39e0-420e-936b-ee5d35288f9c",
  "security_group_id": "d8c81e0f-de6a-4110-8c96-81af3eacb3d1",
  "subnet_id": "0b6cfaea-bce7-48eb-b38d-267c24df5f79",
  "available_zones": [
    "38b0f7a602344246bcb0da47b5d548e7"
  ],
  "product_id": "00300-30308-0--0",
  "kafka_manager_user": "test",
  "kafka_manager_password": "Zxxxx",
  "enable_publicip": true,
  "publicip_id": "87864b85-7097-4c06-9d62-718d7359a503,72c12ba7-fade-4b06-a680-01d335cf786d,11b535df-ed6d-4521-8d00-12bb60beb617",
  "storage_spec_code": "dms.physical.storage.high"
}
```

Response

Response parameters

Table 7-4 describes the parameters.

Table 7-4 Response parameters

Parameter	Type	Description
instance_id	String	Indicates the instance ID.

Example response

```
{
  "instance_id": "8959ab1c-7n1a-yyb1-a05t-93dfc361b32d"
}
```

Status Code

Table 7-5 describes the status code of successful operations. For details about other status codes, see **Status Code**.

Table 7-5 Status code

Status Code	Description
200	The instance is created successfully.

7.1.1.2 Querying an Instance

 **NOTE**

This API is out-of-date and may not be maintained in the future. Please use the API described in **Querying an Instance**.

Function

This API is used to query the details about an instance.

URI

GET /v1.0/{project_id}/instances/{instance_id}

Table 7-6 describes the parameters.

Table 7-6 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
```

Response

Response parameters

[Table 7-7](#) describes the parameters.

Table 7-7 Response parameters

Parameter	Type	Description
name	String	Indicates the instance name.
engine	String	Indicates the message engine.
engine_version	String	Indicates the version of the message engine.
specification	String	Indicates the instance specification.
storage_space	Integer	Indicates the message storage space. Unit: GB
partition_num	String	Indicates the total number of partitions in a Kafka instance.
used_storage_space	Integer	Indicates the used message storage space. Unit: GB
connect_addresses	String	Indicates the IP address of an instance.
port	Integer	Indicates the port number of an instance.
status	String	Indicates the status of an instance. For details, see Instance Status .
instance_id	String	Indicates the instance ID.
resource_spec_code	String	Indicates the resource specifications identifier. <ul style="list-style-type: none"> dms.instance.kafka.cluster.c3.mini: Kafka instance, 100 MB/s reference bandwidth dms.instance.kafka.cluster.c3.small.2: Kafka instance, 300 MB/s reference bandwidth dms.instance.kafka.cluster.c3.middle.2: Kafka instance, 600 MB/s reference bandwidth dms.instance.kafka.cluster.c3.high.2: Kafka instance, 1200 MB/s reference bandwidth

Parameter	Type	Description
type	String	Indicates the instance type. Value: cluster
charging_mode	Integer	Indicates the billing mode. 1 : pay-per-use mode; 0 : yearly/monthly billing.
vpc_id	String	Indicates the ID of a VPC.
vpc_name	String	Indicates the name of a VPC.
created_at	String	Indicates the time when an instance is created. The time is in the format of timestamp, that is, the offset milliseconds from 1970-01-01 00:00:00 UTC to the specified time.
product_id	String	Indicates the product ID.
security_group_id	String	Indicates the security group ID.
security_group_name	String	Indicates the security group name.
subnet_id	String	Indicates the subnet ID.
subnet_name	String	Indicates the subnet name.
subnet_cidr	String	Indicates the subnet CIDR block.
available_zones	Array	Indicates the ID of the AZ to which the instance node belongs. The AZ ID is returned.
user_id	String	Indicates the user ID.
user_name	String	Indicates the username.
access_user	String	Indicates the username of an instance.
order_id	String	Indicates the order ID.
maintain_begin	String	Indicates the time at which a maintenance time window starts. Format: HH:mm:ss
maintain_end	String	Indicates the time at which a maintenance time window ends. Format: HH:mm:ss
enable_publicip	Boolean	Indicates whether to enable public access for an instance. <ul style="list-style-type: none">● true: enable● false: disable
management_connect_address	String	Indicates the connection address of the Kafka Manager of a Kafka instance.

Parameter	Type	Description
ssl_enable	Boolean	Indicates whether to enable security authentication. <ul style="list-style-type: none"> • true: enable • false: disable
enterprise_project_id	String	Indicates the enterprise project ID.
is_logical_volume	Boolean	Distinguishes old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> • true: New instance, which allows dynamic disk capacity expansion without restarting the instance. • false: Old instance.
extend_times	Integer	Indicates the number of disk expansion times. If it exceeds 20, the disk cannot be expanded.
enable_auto_topic	Boolean	Indicates whether automatic topic creation is enabled. <ul style="list-style-type: none"> • true: enabled • false: disabled
total_storage_space	Integer	Indicates the message storage space. Unit: GB
storage_resource_id	String	Indicates the storage resource ID.
storage_spec_code	String	Indicates the I/O specification.
service_type	String	Indicates the service type.
storage_type	String	Indicates the storage type.
retention_policy	String	Indicates the message retention policy.
kafka_public_status	String	Indicates whether Kafka public access is enabled.
public_bandwidth	Integer	Indicates the public network bandwidth.
public_connect_address	String	Indicates the instance IP address for public access. This parameter is displayed only when public access is enabled.
kafka_manager_user	String	Indicates the username for logging in to Kafka Manager.
enable_log_collection	Boolean	Indicates whether log collection is enabled.

Parameter	Type	Description
cross_vpc_info	String	Indicates cross-VPC access information.
ipv6_enable	Boolean	Indicates whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	Indicates the IPv6 connection address.
rest_enable	Boolean	Indicates whether the Kafka REST function is enabled.
rest_connect_address	String	Indicates the Kafka REST connection address.
message_query_inst_enable	Boolean	Indicates whether message query is enabled.
vpc_client_plain	Boolean	Indicates whether intra-VPC plaintext access is enabled.
support_features	String	Indicates the list of features supported by the Kafka instance.
trace_enable	Boolean	Indicates whether message tracing is enabled.
pod_connect_address	String	Indicates the connection address on the tenant side.
disk_encrypted	Boolean	Indicates whether disk encryption is enabled. <ul style="list-style-type: none"> • true: enabled • false: disabled
kafka_private_connect_address	String	Indicates the private connection address of a Kafka instance.
ces_version	String	Indicates the Cloud Eye version.
tags	Array<Object>	Indicates the list of tags.

Table 7-8 tags

Parameter	Type	Description
key	String	Indicates the tag key.
value	String	Indicates the tag value.

Example response

```
{
  "name": "kafka-l00230526",
  "engine": "kafka",
  "port": 9092,
```

```
"status": "RUNNING",
"type": "cluster",
"specification": "100MB",
"engine_version": "XXX",
"connect_address": "192.168.1.116,192.168.1.152,192.168.1.78",
"connect_dn": "",
"instance_id": "ef84dd5f-3ece-4336-8c99-987defd62e3a",
"resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
"charging_mode": 1,
"vpc_id": "2477879f-aebf-496f-a08a-67812885ce9b",
"vpc_name": "vpc-y00502467",
"created_at": "1568797295209",
"product_id": "00300-30308-0--0",
"security_group_id": "008a08e2-10cc-4d9b-90ab-3f3b8f6c3333",
"security_group_name": "z00417080-cce-node-na7j",
"subnet_id": "5ca08fb7-7522-4d95-9fa5-ff6b3592a29d",
"subnet_name": "subnet-cyd-6102",
"subnet_cidr": "192.168.1.0/24",
"available_zones": [
  "ae04cf9d61544df3806a3feeb401b204"
],
"user_id": "2b4af4428ec840dfa1f0f1a32e965567",
"user_name": "laiyh",
"kafka_manager_user": "root",
"maintain_begin": "22:00:00",
"maintain_end": "02:00:00",
"storage_space": 492,
"total_storage_space": 600,
"used_storage_space": 25,
"partition_num": "300",
"ssl_enable": false,
"management_connect_address": "https://192.168.1.116:9999",
"storage_resource_id": "81982562-ce8b-490a-95fa-2b225c292271",
"storage_spec_code": "dms.physical.storage.ultra",
"service_type": "advanced",
"storage_type": "hec",
"enterprise_project_id": "0",
"is_logical_volume": true,
"extend_times": 0,
"retention_policy": "produce_reject",
"ipv6_enable": false,
"ipv6_connect_addresses": [],
"connector_enable": false,
"connector_id": "",
"rest_enable": false,
"rest_connect_address": "",
"message_query_inst_enable": true,
"vpc_client_plain": false,
"support_features":
"feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.
kafka.pulbic.dynamic,feature.physerver.kafka.user.manager",
"trace_enable": false,
"agent_enable": false,
"pod_connect_address": "100.113.16.105:9100,100.113.5.197:9100,100.113.15.231:9100",
"disk_encrypted": false,
"enable_auto_topic": true
}
```

Status Code

Table 7-9 describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-9 Status code

Status Code	Description
200	Specified instance queried successfully.

7.1.1.3 Modifying an Instance

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Modifying Instance Information](#).

Function

This API is used to modify the instance information, including the instance name, description, maintenance window, and security group.

URI

PUT /v1.0/{project_id}/instances/{instance_id}

Table 7-10 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

[Table 7-11](#) describes the parameters.

Table 7-11 Request parameters

Parameter	Type	Mandatory	Description
name	String	No	Indicates the instance name. An instance name consists of 4 to 64 characters including letters, digits, and hyphens (-) and must start with a letter.

Parameter	Type	Mandatory	Description
description	String	No	<p>Indicates the description of an instance. It is a character string containing not more than 1024 characters.</p> <p>NOTE The backslash (\) and quotation mark (") are special characters for JSON packets. When using these characters in a parameter value, add the escape character (\) before these characters, for example, \\ and \".</p>
maintain_begin	String	No	<p>Indicates the time at which a maintenance time window starts.</p> <p>Format: HH:mm:ss</p> <ul style="list-style-type: none"> The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window. For details about how to query the time segments of supported maintenance time windows, see Querying Maintenance Time Windows. The start time must be set to 22:00:00, 02:00:00, 06:00:00, 10:00:00, 14:00:00, or 18:00:00. Parameters maintain_begin and maintain_end must be set in pairs. If parameter maintain_begin is left blank, parameter maintain_end is also left blank. In this case, the system automatically sets the start time to 02:00:00.

Parameter	Type	Mandatory	Description
maintain_end	String	No	<p>Indicates the time at which a maintenance time window ends.</p> <p>Format: HH:mm:ss</p> <ul style="list-style-type: none"> The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window. For details about how to query the time segments of supported maintenance time windows, see Querying Maintenance Time Windows. The end time is four hours later than the start time. For example, if the start time is 22:00:00, the end time is 02:00:00. Parameters maintain_begin and maintain_end must be set in pairs. If parameter maintain_end is left blank, parameter maintain_start is also left blank. In this case, the system automatically sets the end time to 06:00:00.
security_group_id	String	No	Indicates the security group ID.
retention_policy	String	No	<p>Indicates the capacity threshold policy.</p> <p>Options:</p> <ul style="list-style-type: none"> produce_reject: New messages cannot be created. time_base: The earliest messages are deleted.
enterprise_project_id	String	No	Indicates the enterprise project ID.

Example request

Example 1:

```
PUT https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
{
  "name": "dms002",
  "description": "instance description"
}
```

Example 2:

```
PUT https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
{
  "name": "dms002",
  "description": "instance description",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00"
}
```

Response

Response parameters

None.

Example response

None.

Status Code

Table 7-12 describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-12 Status code

Status Code	Description
204	The instance is modified successfully.

7.1.1.4 Deleting an Instance

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Deleting an Instance](#).

Function

This API is used to delete an instance to release all the resources occupied by it.

URI

DELETE /v1.0/{project_id}/instances/{instance_id}

Table 7-13 describes the parameters.

Table 7-13 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

Parameter	Type	Mandatory	Description
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

None.

Example request

```
DELETE https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}
```

Response

Response parameters

None.

Example response

None.

Status Code

Table 7-14 describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-14 Status code

Status Code	Description
204	The instance is deleted successfully.

7.1.1.5 Restarting or Deleting Instances in Batches

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Batch Restarting or Deleting Instances](#).

Function

This API is used to restart or delete instances in batches.

When an instance is being restarted, message retrieval and creation requests of the client will be rejected.

Deleting an instance will delete the data in the instance without any backup. Exercise caution when performing this operation.

URI

POST /v1.0/{project_id}/instances/action

[Table 7-15](#) describes the parameters.

Table 7-15 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

Request

Request parameters

[Table 7-16](#) describes the parameters.

Table 7-16 Request parameters

Parameter	Type	Mandatory	Description
action	String	Yes	Indicates the operation to be performed on instances. The value of this parameter can be restart or delete .
instances	Array	No	Indicates the list of instance IDs.
allFailure	String	No	When set to kafka , indicates all Kafka instances that fail to be created are to be deleted.

Example request

Restarting instances in batches:

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
  "action": "restart",
  "instances": ["54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-dbad-4d79-9610-7163e6f8b640"]
}
```

Deleting instances in batches:

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
  "action": "delete",
  "instances": ["54602a9d-5e22-4239-9123-77e350df4a34", "7166cdea-dbad-4d79-9610-7163e6f8b640"]
}
```

Deleting all instances that fail to be created:

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/action
{
```



```
"action" : "delete",
"allFailure" : "kafka"
}
```

Response

Response parameters

When **action** is set to **delete**, **allFailure** is set to **kafka**, and an empty response is returned, the instances are deleted successfully. [Table 7-17](#) describes the parameters.

Table 7-17 Response parameters

Parameter	Type	Description
results	Array	Indicates the result of instance modification.

Table 7-18 results parameter description

Parameter	Type	Description
instance	String	Indicates the instance ID.
result	String	Indicates an operation result, which can be success or failed

Example response

```
{
  "results": [
    {
      "result": "success",
      "instance": "afc90a2a-a02c-4cba-94d5-58dfa9ad1e0d"
    },
    {
      "result": "success",
      "instance": "67fc5f8d-3986-4f02-bb75-4075a23112de"
    }
  ]
}
```

Status Code

[Table 7-19](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-19 Status code

Status Code	Description
200	The instances are restarted or deleted successfully.
204	Successfully deleting an instance failed to be created.

7.1.1.6 Querying All Instances

 NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Listing All Instances](#).

Function

This API is used to query the instances of a tenant by set conditions.

URI

GET /v1.0/{project_id}/instances?
engine={engine}&name={name}&status={status}&id={id}&includeFailure={includeFailure}&exactMatchName={exactMatchName}&enterprise_project_id={enterprise_project_id}

[Table 7-20](#) describes the parameters.

Table 7-20 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
engine	String	No	Indicates the message engine. Value: kafka
name	String	No	Indicates the instance name.
id	String	No	Indicates the instance ID.
status	String	No	Indicates the instance status. For details, see Instance Status .
includeFailure	String	No	Indicates whether to return the number of instances that fail to be created. If the value is true , the number of instances that failed to be created is returned. If the value is not true , the number is not returned.

Parameter	Type	Mandatory	Description
exactMatchName	String	No	Indicates whether to search for the instance that precisely matches a specified instance name. The default value is false , indicating that a fuzzy search is performed based on a specified instance name. If the value is true , the instance that precisely matches a specified instance name is queried.
enterprise_project_id	String	No	Indicates the enterprise project ID.

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances?
start=1&limit=10&name=&status=&id=&includeFailure=true&exactMatchName=false
```

Response

Response parameters

[Table 7-21](#) describes the parameters.

Table 7-21 Response parameters

Parameter	Type	Description
instances	Array	Indicates instance details.
instance_num	Integer	Indicates the number of instances.

Table 7-22 instance parameter description

Parameter	Type	Description
name	String	Indicates the instance name.
engine	String	Indicates the message engine.
engine_version	String	Indicates the engine version.

Parameter	Type	Description
specification	String	Indicates the specifications of an instance.
storage_space	Integer	Indicates the message storage space. Unit: GB
partition_num	String	Indicates the maximum number of topics in a Kafka instance.
used_storage_space	Integer	Indicates the used message storage space. Unit: GB
connect_addresses	String	Indicates the IP address of an instance.
port	Integer	Indicates the port number of an instance.
status	String	Indicates the status of an instance. For details, see Instance Status .
instance_id	String	Indicates the instance ID.
resource_spec_code	String	Indicates the resource specifications identifier. <ul style="list-style-type: none"> dms.instance.kafka.cluster.c3.mini: Kafka instance, 100 MB/s reference bandwidth dms.instance.kafka.cluster.c3.small.2: Kafka instance, 300 MB/s reference bandwidth dms.instance.kafka.cluster.c3.middle.2: Kafka instance, 600 MB reference bandwidth dms.instance.kafka.cluster.c3.high.2: Kafka instance, 1200 MB reference bandwidth
charging_mode	Integer	Indicates the billing mode. Options: 1: pay-per-use; 0: yearly/monthly
vpc_id	String	Indicates the ID of a VPC.
vpc_name	String	Indicates the name of a VPC.
created_at	String	Indicates the time when an instance is created. The time is in the format of timestamp, that is, the offset milliseconds from 1970-01-01 00:00:00 UTC to the specified time.
user_id	String	Indicates the user ID.
user_name	String	Indicates the username.
access_user	String	Indicates the username of an instance.
order_id	String	Indicates the order ID.
maintain_begin	String	Indicates the time at which a maintenance time window starts. Format: HH:mm:ss

Parameter	Type	Description
maintain_end	String	Time at which the maintenance time window ends. Format: HH:mm:ss
enable_publicip	Boolean	Indicates whether to enable public access for an instance. <ul style="list-style-type: none"> • true: enable • false: disable
management_connect_address	String	Indicates the connection address of the Kafka Manager of a Kafka instance.
ssl_enable	Boolean	Indicates whether to enable security authentication. <ul style="list-style-type: none"> • true: enable • false: disable
enterprise_project_id	String	Indicates the enterprise project ID.
is_logical_volume	Boolean	Distinguishes old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> • true: New instance, which allows dynamic disk capacity expansion without restarting the instance. • false: Old instance.
extend_times	Integer	Indicates the number of disk expansion times. If it exceeds 20, the disk cannot be expanded.
enable_auto_topic	Boolean	Indicates whether automatic topic creation is enabled. <ul style="list-style-type: none"> • true: enabled • false: disabled
type	String	Indicates the instance type. Value: cluster .
product_id	String	Indicates the product ID.
security_group_id	String	Indicates the security group ID.
security_group_name	String	Indicates the security group name.
subnet_id	String	Indicates the subnet ID.
available_zones	Array	Indicates the AZ to which the instance node belongs. The AZ ID is returned.
total_storage_space	Integer	Indicates the message storage space. Unit: GB

Parameter	Type	Description
public_connect_address	String	Indicates the instance IP address for public access. This parameter is displayed only when public access is enabled.
storage_resource_id	String	Indicates the storage resource ID.
storage_spec_code	String	Indicates the I/O specification.
service_type	String	Indicates the service type.
storage_type	String	Indicates the storage type.
retention_policy	String	Indicates the message retention policy.
kafka_public_status	String	Indicates whether Kafka public access is enabled.
public_bandwidth	Integer	Indicates the public network bandwidth.
kafka_manager_user	String	Indicates the username for logging in to Kafka Manager.
enable_log_collection	Boolean	Indicates whether log collection is enabled.
cross_vpc_info	String	Indicates cross-VPC access information.
ipv6_enable	Boolean	Indicates whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	Indicates the IPv6 connection address.
rest_enable	Boolean	Indicates whether the Kafka REST function is enabled.
rest_connect_address	String	Indicates the Kafka REST address.
message_query_inst_enable	Boolean	Indicates whether message query is enabled.
vpc_client_plain	Boolean	Indicates whether intra-VPC plaintext access is enabled.
support_features	String	Indicates the list of features supported by the Kafka instance.
trace_enable	Boolean	Indicates whether message tracing is enabled.
pod_connect_address	String	Indicates the connection address on the tenant side.
disk_encrypted	Boolean	Indicates whether disk encryption is enabled.

Parameter	Type	Description
kafka_private_connect_address	String	Indicates the private connection address of a Kafka instance.
ces_version	String	Indicates the Cloud Eye version.
tags	Array<Object>	Indicates the list of tags.

Table 7-23 tags

Parameter	Type	Description
key	String	Indicates the tag key.
value	String	Indicates the tag value.

Example response

```
{
  "instances": [
    {
      "name": "kafka-l00230526",
      "engine": "kafka",
      "port": 9092,
      "status": "RUNNING",
      "type": "cluster",
      "specification": "100MB",
      "engine_version": "XXX",
      "connect_address": "192.168.1.116,192.168.1.152,192.168.1.78",
      "instance_id": "ef84dd5f-3ece-4336-8c99-987defd62e3a",
      "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
      "charging_mode": 1,
      "vpc_id": "2477879f-aebf-496f-a08a-67812885ce9b",
      "vpc_name": "vpc-y00502467",
      "created_at": "1568797295209",
      "product_id": "00300-30308-0--0",
      "security_group_id": "008a08e2-10cc-4d9b-90ab-3f3b8f6c3333",
      "security_group_name": "z00417080-cce-node-na7j",
      "subnet_id": "5ca08fb7-7522-4d95-9fa5-ff6b3592a29d",
      "available_zones": [
        "ae04cf9d61544df3806a3feeb401b204"
      ],
      "user_id": "2b4af4428ec840dfa1f0f1a32e965567",
      "user_name": "laiyh",
      "kafka_manager_user": "root",
      "maintain_begin": "22:00",
      "maintain_end": "02:00",
      "storage_space": 492,
      "total_storage_space": 600,
      "used_storage_space": 25,
      "partition_num": "300",
      "ssl_enable": false,
      "management_connect_address": "https://192.168.1.116:9999",
      "storage_resource_id": "81982562-ce8b-490a-95fa-2b225c292271",
      "storage_spec_code": "dms.physical.storage.ultra",
      "service_type": "advanced",
      "storage_type": "hec",
      "enterprise_project_id": "0",
      "is_logical_volume": true,
    }
  ]
}
```

```

    "extend_times": 0,
    "retention_policy": "produce_reject",
    "ipv6_enable": false,
    "ipv6_connect_addresses": [],
    "rest_enable": false,
    "rest_connect_address": "",
    "message_query_inst_enable": true,
    "vpc_client_plain": false,
    "support_features":
"feature.physerver.kafka.topic.accesspolicy,message_trace_enable,features.pod.token.access,feature.physerver.
kafka.pulbic.dynamic,feature.physerver.kafka.user.manager",
    "trace_enable": false,
    "agent_enable": false,
    "pod_connect_address": "100.113.16.105:9100,100.113.5.197:9100,100.113.15.231:9100",
    "disk_encrypted": false,
    "enable_auto_topic": true
  }
],
  "instance_num": 1
}

```

Status Code

[Table 7-24](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-24 Status code

Status Code	Description
200	All instances are queried successfully.

7.1.1.7 Creating a Topic in a Kafka Instance

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Creating a Topic for a Kafka Instance](#).

Function

This API is used to create a topic in a Kafka instance.

URI

POST /v1.0/{project_id}/instances/{instance_id}/topics

[Table 7-25](#) describes the parameters.

Table 7-25 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

Parameter	Type	Mandatory	Description
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

[Table 7-26](#) describes the parameter.

Table 7-26 Request parameters

Parameter	Type	Mandatory	Description
id	String	Yes	Indicates the name of a topic. A topic name consists of 4 to 64 characters, starts with a letter, and contains only letters, hyphens (-), underscores (_), and digits.
partition	Integer	No	Indicates the number of topic partitions, which is used to set the number of concurrently consumed messages. Value range: 1–100. Default value: 3 .
replication	Integer	No	Indicates the number of replicas, which is configured to ensure data reliability. Value range: 1–3. Default value: 3 .
sync_replication	Boolean	No	Indicates whether to enable synchronous replication. After this function is enabled, the acks parameter on the producer client must be set to -1 . Otherwise, this parameter does not take effect. By default, synchronous replication is disabled.
retention_time	Integer	No	Indicates the retention period of a message. Its default value is 72 . Value range: 1–720. Unit: hour.
sync_message_flush	Boolean	No	Indicates whether to enable synchronous flushing. Default value: false . Synchronous flushing compromises performance.

Example request

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics
{
  "id" : "haha",
  "partition" : 3,
  "replication" : 3,
  "sync_replication " : true,
  "retention_time" : 10,
  "sync_message_flush" : true
}
```

Response

Response parameters

[Table 7-27](#) describes the parameter.

Table 7-27 Response parameters

Parameter	Type	Description
id	String	Indicates the name of a topic.

Example response

```
{
  "id": "haha"
}
```

Status Code

[Table 7-28](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-28 Status code

Status Code	Description
200	The topic is created successfully.

7.1.1.8 Querying a Topic in a Kafka Instance

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Listing Topics of a Kafka Instance](#).

Function

This API is used to query details about a topic in a Kafka instance.

URI

GET /v1.0/{project_id}/instances/{instance_id}/topics

[Table 7-29](#) describes the parameter.

Table 7-29 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics
```

Response

Response parameters

[Table 7-30](#) describes the response parameter.

Table 7-30 Response parameter

Parameter	Type	Description
total	Integer	Indicates the total number of topics.
size	Integer	Indicates the maximum number of records to be displayed on a page.
remain_partitions	Integer	Indicates the number of remaining partitions.
max_partitions	Integer	Indicates the total number of partitions.
topics	Array	Indicates the list of topics.

Table 7-31 Parameter description

Parameter	Type	Description
policiesOnly	Boolean	Whether this policy is the default policy.
id	String	Indicates the topic name.

Parameter	Type	Description
replication	Integer	Indicates the number of replicas, which is configured to ensure data reliability.
partition	Integer	Indicates the number of topic partitions, which is used to set the number of concurrently consumed messages.
retention_time	Integer	Indicates the retention period of a message.
sync_replication	Boolean	Indicates whether to enable synchronous replication. After this function is enabled, the acks parameter on the producer client must be set to -1 . Otherwise, this parameter does not take effect. By default, synchronous replication is disabled.
sync_message_flush	Boolean	Indicates whether to enable synchronous flushing. Synchronous flushing compromises performance.
external_configs	Object	Indicates the extended configuration.
topic_type	Integer	Indicates the topic type.

Example response

```
{
  "count": 1,
  "topics": [
    {
      "id": "topic-test",
      "replication": 3,
      "partition": 4,
      "retention_time": 72,
      "sync_replication": "false",
      "sync_message_flush": "false"
    }
  ]
}
```

Status Code

[Table 7-32](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-32 Status code

Status Code	Description
200	The information is queried successfully.

7.1.1.9 Deleting Topics in a Kafka Instance in Batches

 **NOTE**

This API is out-of-date and may not be maintained in the future. Please use the API described in [Batch Deleting Topics of a Kafka Instance](#).

Function

This API is used to delete topics in a Kafka instance in batches.

URI

POST /v1.0/{project_id}/instances/{instance_id}/topics/delete

[Table 7-33](#) describes the parameter.

Table 7-33 Parameters

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
instance_id	String	Yes	Indicates the instance ID.

Request

Request parameters

[Table 7-34](#) describes the parameter.

Table 7-34 Request parameter

Parameter	Type	Mandatory	Description
topics	Array	Yes	Indicates the list of topics to be deleted.

Example request

```
POST https://{dms_endpoint}/v1.0/{project_id}/instances/{instance_id}/topics/delete
{
  "topics": ["hah", "aabb"]
}
```

Response

Response parameters

Table 7-35 describes the parameter.

Table 7-35 Response parameters

Parameter	Type	Description
topics	Array	Indicates the list of topics.

Table 7-36 topics parameter description

Parameter	Type	Description
id	String	Indicates the topic name.
success	Boolean	Indicates whether the topics are deleted.

Example response

```
{
  "topics": [{
    "id": "haha",
    "success": true
  }, {
    "id": "aabb",
    "success": true
  }
]
```

Status Code

Table 7-37 describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-37 Status code

Status Code	Description
200	The topics are successfully deleted.

7.1.2 Other APIs

7.1.2.1 Querying AZ Information

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Listing AZ Information](#).

Function

This API is used to query the AZ ID.

URI

GET /v1.0/availableZones

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/availableZones
```

Response

Response parameters

[Table 7-38](#) and [Table 7-39](#) describe the parameters.

Table 7-38 Response parameters

Parameter	Type	Description
region_id	String	Indicates the region ID.
available_zones	Array	Indicates details of AZs. For details, see Table 7-39 .

Table 7-39 available_zones parameter description

Parameter	Type	Description
soldOut	Boolean	Indicates whether resources are sold out.
id	String	Indicates the ID of an AZ.
code	String	Indicates the code of an AZ.
name	String	Indicates the name of an AZ.
port	String	Indicates the port number of an AZ.
resource_availability	String	Indicates whether an AZ has available resources. <ul style="list-style-type: none">● true: The AZ has available resources.● false: Resources of the AZ have been sold out.

Example response

```
{
  regionId: "XXXX",
  available_zones:[
    {
      "id":"1d7b939b382c4c3bb3481a8ca10da768",
      "name":"az10.dc1",
      "code":"az10.dc1",
      "port":"8002",
      "resource_availability": "true"
    },
    {
      "id":"1d7b939b382c4c3bb3481a8ca10da769",
      "name":"az10.dc2",
      "code":"az10.dc2",
      "port":"8002",
      "resource_availability": "true"
    }
  ]
}
```

Status Code

[Table 7-40](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-40 Status code

Status Code	Description
200	The AZ information is successfully queried.

7.1.2.2 Querying Product Specifications

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Querying Product Specifications](#).

Function

This API is used to query the product specifications to configure the product ID.

For example, to create a pay-per-use Kafka instance with 100 MB/s bandwidth, locate the section where the value of **bandwidth** is **100MB** under "Hourly" in the response message. Then, the value of **product_id** in the same section is the product ID that should be configured for the Kafka instance.

unavailable_zones indicates unavailable AZs. If the value is empty, all AZs are available. Otherwise, the AZs listed in the value do not have sufficient resources. Ensure that the AZs where you want to create the instance are not listed here.

URI

GET /v1.0/products?engine={engine}

[Table 7-41](#) describes the parameter.

Table 7-41 Parameters

Parameter	Type	Mandatory	Description
engine	String	Yes	Indicates the message engine.

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/products?engine={engine}
```

Response

Response parameters

[Table 7-43](#) describes the response parameters.

Table 7-42 Parameters

Parameter	Type	Description
Hourly	Array	Indicates the list of pay-per-use products.
Monthly	Array	Indicates the list of products in yearly/monthly billing mode. Currently, you cannot create yearly/monthly Kafka instances by calling an API.

Table 7-43 Parameters

Parameter	Type	Description
name	String	Indicates the message engine, which is kafka .
version	String	Version of the message engine.
values	Array	Indicates product specifications. For details, see Table 7-44 .

Table 7-44 values parameter description

Parameter	Type	Description
detail	Array	Indicates the details of specifications. For details, see Table 7-45 .

Parameter	Type	Description
name	String	Indicates the instance type.
unavailable_zones	Array	Indicates AZs where resources are sold out.
available_zones	Array	Indicates AZs where there are available resources.

Table 7-45 detail parameter description

Parameter	Type	Description
tps	String	Indicates the maximum number of messages per unit time.
storage	String	Indicates the message storage space.
partition_num	String	Indicates the maximum number of topics in a Kafka instance.
product_id	String	Indicates the product ID.
spec_code	String	Indicates the specification ID.
io	Array	Indicates the I/O information. For details, see Table 7-46 .
bandwidth	String	Indicates the bandwidth of a Kafka instance.
available_zones	Array	Indicates AZs where there are available resources.
ecs_flavor_id	String	Indicates the flavors of the corresponding ECS.
arch_type	String	Indicates the instance architecture type. Currently, only x86 is supported.

Table 7-46 io parameter description

Parameter	Type	Description
io_type	String	Indicates the I/O type.
storage_spec_code	String	Indicates the I/O specification.
available_zones	Array	Indicates AZs where there are available I/O resources.
unavailable_zones	Array of strings	Indicates AZs where I/O resources are sold out.

Parameter	Type	Description
volume_type	String	Indicates the disk type.

Example response

```
{
  "Hourly": [{
    "name": "kafka",
    "version": "XXX",
    "values": [{
      "detail": [{
        "tps": "50000",
        "storage": "600",
        "partition_num": "300",
        "product_id": "00300-30308-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.mini",
        "io": [{
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high",
          "volume_type": "SAS"
        }],
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra",
          "volume_type": "SSD"
        }
      ]],
      "bandwidth": "100MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.large.2"
    },
    {
      "tps": "100000",
      "storage": "1200",
      "partition_num": "900",
      "product_id": "00300-30310-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.small.2",
      "io": [{
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high",
        "volume_type": "SAS"
      }],
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }
    ]],
      "bandwidth": "300MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.xlarge.2"
    },
    {
      "tps": "200000",
      "storage": "2400",
      "partition_num": "1800",
      "product_id": "00300-30312-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.middle.2",
      "io": [{
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }],
      "bandwidth": "600MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.2xlarge.2"
    }
  ]
}
```

```

    {
      "tps": "300000",
      "storage": "4800",
      "partition_num": "1800",
      "product_id": "00300-30314-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.high.2",
      "io": [{
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }],
      "bandwidth": "1200MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3ne.2xlarge.2"
    },
    {
      "name": "cluster",
      "unavailable_zones": []
    }
  ]],
  "Monthly": [{
    "name": "kafka",
    "version": "XXX",
    "values": [{
      "detail": [{
        "tps": "50000",
        "storage": "600",
        "partition_num": "300",
        "product_id": "00300-30309-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.mini",
        "io": [{
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high",
          "volume_type": "SAS"
        }],
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra",
          "volume_type": "SSD"
        }
      ]],
      "bandwidth": "100MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.large.2"
    },
    {
      "tps": "100000",
      "storage": "1200",
      "partition_num": "900",
      "product_id": "00300-30311-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.small.2",
      "io": [{
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high",
        "volume_type": "SAS"
      }],
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra",
        "volume_type": "SSD"
      }
    ]],
      "bandwidth": "300MB",
      "unavailable_zones": [],
      "ecs_flavor_id": "c3.xlarge.2"
    },
    {
      "tps": "200000",
      "storage": "2400",
      "partition_num": "1800",
      "product_id": "00300-30313-0--0",
      "spec_code": "dms.instance.kafka.cluster.c3.middle.2",

```

```

        "io": [{
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }],
        "bandwidth": "600MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3.2xlarge.2"
    },
    {
        "tps": "300000",
        "storage": "4800",
        "partition_num": "1800",
        "product_id": "00300-30315-0--0",
        "spec_code": "dms.instance.kafka.cluster.c3.high.2",
        "io": [{
            "io_type": "ultra",
            "storage_spec_code": "dms.physical.storage.ultra",
            "volume_type": "SSD"
        }],
        "bandwidth": "1200MB",
        "unavailable_zones": [],
        "ecs_flavor_id": "c3ne.2xlarge.2"
    }],
    "name": "cluster",
    "unavailable_zones": []
}
}
}

```

Status Code

Table 7-47 describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-47 Status code

Status Code	Description
200	Product specifications queried successfully.

7.1.2.3 Querying Maintenance Time Windows

 **NOTE**

This API is out-of-date and may not be maintained in the future. Please use the API described in [Listing Maintenance Time Windows](#).

Function

This API is used to query the start and end time of the maintenance window.

URI

GET /v1.0/instances/maintain-windows

Request

Request parameters

None.

Example request

```
GET https://{dms_endpoint}/v1.0/instances/maintain-windows
```

Response

Response parameters

[Table 7-48](#) and [Table 7-49](#) describe the response parameters.

Table 7-48 Response parameters

Parameter	Type	Description
maintain_windows	Array	Indicates a list of supported maintenance time windows.

Table 7-49 maintain_windows parameter description

Parameter	Type	Description
seq	Integer	Indicates the sequential number of a maintenance time window.
begin	String	Indicates the time at which a maintenance time window starts.
end	String	Indicates the time at which a maintenance time window ends.
default	Boolean	Indicates whether a maintenance time window is set to the default time segment.

Example response

```
{
  "maintain_windows": [{
    "default": false,
    "seq": 1,
    "begin": "22:00:00",
    "end": "02:00:00"
  },
  {
    "default": true,
    "seq": 2,
    "begin": "02:00:00",
    "end": "06:00:00"
  }
]
```

```

    "default": false,
    "seq": 3,
    "begin": "06:00:00",
    "end": "10:00:00"
  },
  {
    "default": false,
    "seq": 4,
    "begin": "10:00:00",
    "end": "14:00:00"
  },
  {
    "default": false,
    "seq": 5,
    "begin": "14:00:00",
    "end": "18:00:00"
  },
  {
    "default": false,
    "seq": 6,
    "begin": "18:00:00",
    "end": "22:00:00"
  }
}

```

Status Code

[Table 7-50](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

Table 7-50 Status code

Status Code	Description
200	The maintenance time windows are queried successfully.

7.2 API V2

7.2.1 Managing Lifecycle

7.2.1.1 Querying an Instance

Function

This API is used to query the details about an instance.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Querying an Instance](#).

Call Method

For details, see [How to Call an API](#).

URI

GET /v2/{engine}/{project_id}/instances/{instance_id}

Table 7-51 URI parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Indicates the message engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Indicates the instance ID.

Request

None.

Response

Status code: 200

Table 7-52 Response body parameters

Parameter	Type	Description
name	String	Indicates the instance name.
engine	String	Indicates the message engine.
engine_version	String	Indicates the engine version.
description	String	Indicates the description of the instance.
specification	String	Indicates the instance specification.
storage_space	Integer	Indicates the message storage space. Unit: GB
partition_num	String	Number of partitions in a Kafka instance.
used_storage_space	Integer	Indicates the used message storage space. Unit: GB
dns_enable	Boolean	Whether to enable domain name access to an instance. <ul style="list-style-type: none">● true: enable● false: disable
connect_address	String	Indicates the IP address of an instance.

Parameter	Type	Description
port	Integer	Indicates the port number of an instance.
status	String	Indicates the status of an instance. See Instance statuses .
instance_id	String	Indicates the instance ID.
resource_spec_code	String	Indicates the resource specifications identifier. <ul style="list-style-type: none">• dms.instance.kafka.cluster.c3.mini: Kafka instance, 100 MB/s reference bandwidth• dms.instance.kafka.cluster.c3.small.2: Kafka instance, 300 MB/s reference bandwidth• dms.instance.kafka.cluster.c3.middle.2: Kafka instance, 600 MB/s reference bandwidth• dms.instance.kafka.cluster.c3.high.2: Kafka instance, 1200 MB/s reference bandwidth
charging_mode	Integer	Indicates the billing mode. 1 : pay-per-use mode; 0 : yearly/monthly billing.
vpc_id	String	Indicates the ID of a VPC.
vpc_name	String	Indicates the name of a VPC.
created_at	String	Indicates the time when an instance is created. The time is in the format of timestamp, that is, the offset milliseconds from 1970-01-01 00:00:00 UTC to the specified time.
subnet_name	String	Indicates the subnet name.
subnet_cidr	String	Indicates the subnet CIDR block.
user_id	String	User ID.
user_name	String	Indicates the username.
access_user	String	Username for accessing an instance.
order_id	String	Indicates an order ID. This parameter has a value only when the billing mode is yearly/monthly mode.
maintain_begin	String	Time at which the maintenance time window starts. Format: <i>HH:mm:ss</i> .

Parameter	Type	Description
maintain_end	String	Time at which the maintenance window ends. The format is <i>HH:mm:ss</i> .
enable_publicip	Boolean	Indicates whether to enable public access for an instance. <ul style="list-style-type: none"> • true: enable • false: disable
management_connect_address	String	Indicates the connection address of the Kafka Manager of a Kafka instance.
ssl_enable	Boolean	Indicates whether to enable security authentication. <ul style="list-style-type: none"> • true: enable • false: disable
broker_ssl_enable	Boolean	Indicates whether to enable encrypted replica transmission among brokers. <ul style="list-style-type: none"> • true: enable • false: disable
kafka_security_protocol	String	Security protocol used after SASL is enabled. <ul style="list-style-type: none"> • SASL_SSL: Data is encrypted with SSL certificates for high-security transmission. • SASL_PLAINTEXT: Data is transmitted in plaintext with username and password authentication. This protocol uses the SCRAM-SHA-512 mechanism to deliver better performance.
sasl_enabled_mechanisms	Array of strings	Authentication mechanism used after SASL is enabled. <ul style="list-style-type: none"> • PLAIN: simple username and password verification. • SCRAM-SHA-512: user credential verification, which is more secure than PLAIN.
ssl_two_way_enable	Boolean	Indicates whether to enable two-way authentication.
cert_replaced	Boolean	Whether the certificate can be replaced.
public_management_connect_addresses	String	Address for accessing Kafka Manager over public networks.

Parameter	Type	Description
enterprise_project_id	String	Indicates the enterprise project ID.
is_logical_volume	Boolean	Distinguishes old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> • true: New instance, which allows dynamic disk capacity expansion without restarting the instance. • false: Old instance.
extend_times	Integer	Indicates the number of disk expansion times. If it exceeds 20, the disk cannot be expanded.
enable_auto_topic	Boolean	Indicates whether to enable automatic topic creation. <ul style="list-style-type: none"> • true: enable • false: disabled
type	String	Indicates the instance type. Value: cluster .
product_id	String	Indicates the product ID.
security_group_id	String	Indicates the security group ID.
security_group_name	String	Indicates the security group name.
subnet_id	String	Indicates the subnet ID.
available_zones	Array of strings	Indicates the ID of the AZ to which the instance node belongs. The AZ ID is returned.
available_zone_names	Array of strings	Indicates the AZ name to which the instance node belongs. The AZ name is returned.
total_storage_space	Integer	Indicates the message storage space. Unit: GB
public_connect_address	String	Indicates the instance IP address for public access. This parameter is displayed only when public access is enabled.
public_connect_domain_name	String	Domain name for public access to an instance. This parameter is displayed only when public access is enabled.

Parameter	Type	Description
storage_resource_id	String	Indicates the storage resource ID.
storage_spec_code	String	Indicates the I/O specification.
service_type	String	Indicates the service type.
storage_type	String	Indicates the storage type.
retention_policy	String	Indicates the message retention policy.
kafka_public_status	String	Indicates whether Kafka public access is enabled.
public_bandwidth	Integer	Public network access bandwidth.
enable_log_collection	Boolean	Indicates whether log collection is enabled.
new_auth_cert	Boolean	Indicates whether to enable a new certificate.
cross_vpc_info	String	Indicates cross-VPC access information.
ipv6_enable	Boolean	Indicates whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	Indicates the IPv6 connection address.
connector_enable	Boolean	Indicates whether dumping is enabled. Dumping is not supported for the new specification type.
connector_node_number	Integer	Number of connectors.
connector_id	String	Indicates the ID of the dumping task.
rest_enable	Boolean	Indicates whether the Kafka REST function is enabled.
rest_connect_address	String	Indicates the Kafka REST connection address.
public_boundwidth	Integer	Public network access bandwidth. To be deleted.
message_query_instant_enable	Boolean	Indicates whether message query is enabled.
vpc_client_plain	Boolean	Indicates whether intra-VPC plaintext access is enabled.
support_features	String	Indicates the list of features supported by the Kafka instance.

Parameter	Type	Description
trace_enable	Boolean	Indicates whether message tracing is enabled.
agent_enable	Boolean	Indicates whether the proxy is enabled.
pod_connect_address	String	Indicates the connection address on the tenant side.
disk_encrypted	Boolean	Indicates whether disk encryption is enabled.
disk_encrypted_key	String	Disk encryption key. If disk encryption is not enabled, this parameter is left blank.
kafka_private_connect_address	String	Indicates the private connection address of a Kafka instance.
kafka_private_connect_domain_name	String	Indicates the private domain name of a Kafka instance.
ces_version	String	Indicates the Cloud Eye version.
public_access_enabled	String	Time when public access was enabled for an instance. Values: <ul style="list-style-type: none"> ● true: Public access is enabled. ● actived: Public access is enabled. ● closed: Public access is disabled. ● false: Public access is disabled.
node_num	Integer	Number of nodes.
port_protocols	PortProtocolsEntity object	Connection modes and addresses supported by an instance.
enable_acl	Boolean	Indicates whether access control is enabled.
new_spec_billing_enable	Boolean	Whether billing based on new specifications is enabled.
broker_num	Integer	Node quantity
tags	Array of TagEntity objects	Indicates the list of tags.
dr_enable	Boolean	Indicates whether DR is enabled.

Table 7-53 PortProtocolsEntity

Parameter	Type	Description
private_plain_enable	Boolean	Indicates whether private plaintext access is supported by an instance. <ul style="list-style-type: none"> • true: Yes • false: No
private_plain_address	String	Connection addresses of Kafka private plaintext access.
private_plain_domain_name	String	Private plaintext connection domain name.
private_sasl_ssl_enable	Boolean	Whether the instance supports private SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_ssl_address	String	Connection addresses of the Kafka private SASL_SSL access mode.
private_sasl_ssl_domain_name	String	Private SASL_SSL connection domain name.
private_sasl_plaintext_enable	Boolean	Whether the instance supports private SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
private_sasl_plaintext_address	String	Connection addresses of the Kafka private SASL_PLAINTEXT access mode.
private_sasl_plaintext_domain_name	String	Private SASL_PLAINTEXT connection domain name.
public_plain_enable	Boolean	Whether the instance supports public plaintext access. <ul style="list-style-type: none"> • true: Yes • false: No
public_plain_address	String	Connection addresses of Kafka public plaintext access.
public_plain_domain_name	String	Public plaintext connection domain name.
public_sasl_ssl_enable	Boolean	Whether the instance supports public SASL_SSL access. <ul style="list-style-type: none"> • true: Yes • false: No

Parameter	Type	Description
public_sasl_ssl_address	String	Connection addresses of the Kafka public SASL_SSL access mode.
public_sasl_ssl_domain_name	String	Public SASL_SSL connection domain name.
public_sasl_plaintext_enable	Boolean	Whether the instance supports public SASL_PLAINTEXT access. <ul style="list-style-type: none"> • true: Yes • false: No
public_sasl_plaintext_address	String	Connection addresses of the Kafka public SASL_PLAINTEXT access mode.
public_sasl_plaintext_domain_name	String	Public SASL_PLAINTEXT connection domain name.

Table 7-54 TagEntity

Parameter	Type	Description
key	String	Tag key, which: <ul style="list-style-type: none"> • Cannot be left blank. • Must be unique for the same instance. • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code> • Cannot start with <code>_sys_</code>. • Cannot start or end with a space.
value	String	Tag value. <ul style="list-style-type: none"> • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>._:=-@</code>

Example Request

Querying a Specified Instance.

```
GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}
```

Example Response

Status code: 200

Specified instance queried successfully.

```
{
  "name": "kafka-2085975099",
  "engine": "kafka",
  "port": 9092,
  "status": "RUNNING",
  "type": "cluster",
  "specification": "100MB",
  "engine_version": "1.1.0",
  "connect_address": "192.168.0.100,192.168.0.61,192.168.0.72",
  "instance_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "resource_spec_code": "dms.instance.kafka.cluster.c3.mini",
  "charging_mode": 1,
  "vpc_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "vpc_name": "dms-test",
  "created_at": "1585618587087",
  "product_id": "00300-30308-0--0",
  "security_group_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "security_group_name": "Sys-default",
  "subnet_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "available_zones": [ "38b0f7a602344246bcb0da47b5d548e7" ],
  "available_zone_names": [ "AZ1" ],
  "user_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "user_name": "paas_dms",
  "access_user": "root",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00",
  "enable_log_collection": false,
  "new_auth_cert": false,
  "storage_space": 492,
  "total_storage_space": 600,
  "used_storage_space": 25,
  "partition_num": "300",
  "enable_publicip": false,
  "ssl_enable": false,
  "broker_ssl_enable": false,
  "cert_replaced": false,
  "kafka_security_protocol": "SASL_SSL",
  "management_connect_address": "https://192.168.0.100:9999",
  "cross_vpc_info": "{\n  \"192.168.0.61\":{\n    \"advertised_ip\": \"192.168.0.61\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\", \"192.168.0.72\":{\n      \"advertised_ip\": \"192.168.0.72\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\", \"192.168.0.100\":{\n        \"advertised_ip\": \"192.168.0.100\", \"port\": 9011, \"port_id\": \"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx\"}}}}",
  "storage_resource_id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "storage_spec_code": "dms.physical.storage.ultra",
  "service_type": "advanced",
  "storage_type": "hec",
  "enterprise_project_id": "0",
  "retention_policy": "produce_reject",
  "ipv6_enable": false,
  "ipv6_connect_addresses": [ ],
  "rest_enable": false,
  "rest_connect_address": "",
  "kafka_public_status": "closed",
  "public_bandwidth": 0,
  "trace_enable": false,
  "agent_enable": false,
  "pod_connect_address": "100.86.75.15:9080,100.86.142.77:9080,100.86.250.167:9080",
  "disk_encrypted": false,
  "kafka_private_connect_address": "192.168.0.61:9092,192.168.0.100:9092,192.168.0.72:9092",
  "new_spec_billing_enable": false,
  "ces_version": "linux",
  "port_protocols": "{\n  \"private_plain_enable\": true, \"private_plain_address\": \"192.xxx.xxx.xxx:9092,192.xxx.xxx.xxx:9092,192.xxx.xxx.xxx:9092\", \"private_sasl_ssl_enable\": true, \"private_sasl_ssl_address\": \"192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093,192.xxx.xxx.xxx:9093\", \"private_sasl_plaintext_enable\": false, \"private_sasl_plaintext_address\": \"\", \"public_plain_enable\": true, \"public_plain_address\": \"100.xxx.xxx.xxx:9094,100.xxx.xxx.xxx:9094,100.xxx.xxx.xxx:9094\", \"public_sasl_ssl_enable\": true, \"public_sasl_ssl_address\": \"100.xxx.xxx.xxx:9095,100.xxx.xxx.xxx:9095,100.xxx.xxx.xxx:9095\", \"public_sasl_plaintext_enable\":
```



```

false,\"public_sasl_plaintext_address\": \"\"}
}
    
```

Status Code

Status Code	Description
200	Specified instance queried successfully.

Error Code

See [Error Codes](#).

7.2.2 Instance Management

7.2.2.1 Adding Partitions to a Topic for a Kafka Instance

Function

This API is used to add partitions to a topic for a Kafka instance.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Modifying Topics of a Kafka Instance](#).

URI

POST /v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions-reassignment

Table 7-55 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
topic	Yes	String	Topic name.

Request

Table 7-56 Request parameter

Parameter	Mandatory	Type	Description
partition	No	Integer	Total number of partitions after the addition. The value must be larger than current number of partitions. Maximum value: 100 .

Response

None.

Example Request

Adding partitions to a topic.

```
POST https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/topics/{topic}/partitions-reassignment
```

```
{
  "partition" : 3
}
```

Example Response

None.

Status Code

Status Code	Description
204	Partitions added successfully.

Error Code

For details, see [Error Codes](#).

7.2.3 Managing Consumer Groups

7.2.3.1 Resetting Consumer Group Offset to the Specified Position

Function

Kafka instances do not support resetting the consumer offset online. Before resetting, stop the client for which the offset is to be reset. After a client is

stopped, the server considers the client offline only after the time period specified in `ConsumerConfig.SESSION_TIMEOUT_MS_CONFIG` (1000 ms by default).

 **NOTE**

This API is out-of-date and may not be maintained in the future. Use the API described in [Resetting Consumer Group Offset to the Specified Position](#).

Call Method

For details, see [How to Call an API](#).

URI

POST `/v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset`

Table 7-57 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.
group	Yes	String	Consumer group name.

Request Parameters

Table 7-58 Request body parameters

Parameter	Mandatory	Type	Description
topic	No	String	Topic name.
partition	Yes	Integer	Partition number. The default value is <code>-1</code> , indicating that all partitions are reset.

Parameter	Mandatory	Type	Description
message_offset	No	Long	<p>This parameter resets consumer group offset to the specified position.</p> <ul style="list-style-type: none"> • If this position is earlier than the current earliest offset, the offset will be reset to the earliest position. • If this position is later than the current latest offset, the offset will be reset to the latest position. <p>Either message_offset or timestamp must be specified.</p>
timestamp	No	Long	<p>This parameter resets consumer group offset to the specified time. The value is a UNIX timestamp, in millisecond.</p> <ul style="list-style-type: none"> • If this timestamp is earlier than the current earliest timestamp, the offset will be reset to the earliest timestamp. • If this timestamp is later than the current latest timestamp, the offset will be reset to the latest timestamp. <p>Either message_offset or timestamp must be specified.</p>

Response Parameters

None

Example Requests

- A consumer group offset is reset to the specified position.
 POST `https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset`

```
{
  "topic" : "test",
  "partition" : 0,
  "message_offset" : 10
}
```

- A consumer group offset is reset to the specified time.
POST `https://{endpoint}/v2/{project_id}/instances/{instance_id}/management/groups/{group}/reset-message-offset`

```
{
  "topic" : "test",
  "partition" : 0,
  "timestamp" : 1571812144000
}
```

Example Response

None

SDK Code Samples

The following are SDK code samples.

Java

- A consumer group offset is reset to the specified position.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetMessageOffsetRequest request = new ResetMessageOffsetRequest();
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();
        body.withMessageOffset(10L);
        body.withPartition(0);
        body.withTopic("test");
        request.withBody(body);
        try {
            ResetMessageOffsetResponse response = client.resetMessageOffset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

- A consumer group offset is reset to the specified time.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ResetMessageOffsetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ResetMessageOffsetRequest request = new ResetMessageOffsetRequest();
        ResetMessageOffsetReq body = new ResetMessageOffsetReq();
        body.withTimestamp(1571812144000L);
        body.withPartition(0);
        body.withTopic("test");
        request.withBody(body);
        try {
            ResetMessageOffsetResponse response = client.resetMessageOffset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

- A consumer group offset is reset to the specified position.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetRequest()
        request.body = ResetMessageOffsetReq(
            message_offset=10,
            partition=0,
            topic="test"
        )
        response = client.reset_message_offset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- A consumer group offset is reset to the specified time.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ResetMessageOffsetRequest()
        request.body = ResetMessageOffsetReq(
            timestamp=1571812144000,
            partition=0,
            topic="test"
        )
```

```
)
response = client.reset_message_offset(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- A consumer group offset is reset to the specified position.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetRequest{}
    messageOffsetResetMessageOffsetReq:= int64(10)
    topicResetMessageOffsetReq:= "test"
    request.Body = &model.ResetMessageOffsetReq{
        MessageOffset: &messageOffsetResetMessageOffsetReq,
        Partition: int32(0),
        Topic: &topicResetMessageOffsetReq,
    }
    response, err := client.ResetMessageOffset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- A consumer group offset is reset to the specified time.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)


```



```

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetMessageOffsetRequest{
        timestampResetMessageOffsetReq:= int64(1571812144000)
        topicResetMessageOffsetReq:= "test"
        request.Body = &model.ResetMessageOffsetReq{
            Timestamp: &timestampResetMessageOffsetReq,
            Partition: int32(0),
            Topic: &topicResetMessageOffsetReq,
        }
    }
    response, err := client.ResetMessageOffset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

Status Codes

Status Code	Description
204	Successfully reset the consumer group offset to the specified position.

Error Codes

See [Error Codes](#).

7.2.3.2 Querying All Consumer Groups

Function

This API is used to query all consumer groups.

Call Method

For details, see [How to Call an API](#).

URI

GET /v2/{engine}/{project_id}instances/{instance_id}/groups

Table 7-59 URI parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Indicates the message engine.
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
instance_id	Yes	String	Instance ID.

Table 7-60 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset, which is the position where the query starts. The value must be greater than or equal to 0.
limit	No	Integer	Maximum number of consumer group IDs returned in the current query. The default value is 10. The value ranges from 1 to 50.
group	No	String	Filtering consumer group names that contain specific keywords.
topic	No	String	If a topic is specified, only the consumer groups of the topic are queried.

Request

None.

Response

Status code: 200

Table 7-61 Response body parameters

Parameter	Type	Description
groups	Array of GroupInfoSimple objects	All consumer groups
total	Integer	Total number of consumer groups

Table 7-62 GroupInfoSimple

Parameter	Type	Description
createdAt	Long	Creation time
group_id	String	Consumer group ID
state	String	Consumer group status. The value can be: <ul style="list-style-type: none"> • Dead: The consumer group has no members and no metadata. • Empty: The consumer group has metadata but has no member. • PreparingRebalance: The consumer group is to be rebalanced. • CompletingRebalance: All members have joined the group. • Stable: Members in the consumption group can consume messages normally.
coordinator_id	Integer	Coordinator ID
group_desc	String	Consumer group description
lag	Long	Accumulated messages

Example Request

```
GET https://{endpoint}/v2/{engine}/{project_id}/instances/{instance_id}/groups
```

Example Response

Status code: 200

All consumer groups are queried successfully.

```
{
  "groups": [ {
    "createdAt": 1691401194847,
    "group_id": "consumer-1",
```

```

"state" : "EMPTY",
"coordinator_id" : 1,
"lag" : 0,
"group_desc" : null
}, {
"createdAt" : 1691401194960,
"group_id" : "consumer-2",
"state" : "STABLE",
"coordinator_id" : 2,
"lag" : 0,
"group_desc" : null
}, {
"createdAt" : 1691401207309,
"group_id" : "consumer-3",
"state" : "STABLE",
"coordinator_id" : 3,
"lag" : 0,
"group_desc" : null
} ],
"total" : 3
}

```

Status Code

Status Code	Description
200	All consumer groups are queried successfully.

Error Code

See [Error Codes](#).

7.2.4 Smart Connect

7.2.4.1 Modifying Dumping Task Quotas

Function

This API is used to modify dumping task quotas.

You can call this API to increase the dumping task quota for instances created before September 2022. For instances created in September 2022 or later, the dumping task quota is already the maximum by default. An error is reported when you call this API to modify the dumping task quota, because the dumping task quota cannot be reduced.

Call Method

For details, see [How to Call an API](#).

URI

PUT /v2/{project_id}/connectors/{connector_id}/sink-tasks

Table 7-63 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
connector_id	Yes	String	Instance dump ID. For details, see Querying Instance IDs .

Request Parameters

Table 7-64 Request body parameters

Parameter	Mandatory	Type	Description
sink_max_tasks	Yes	Integer	Total number of dump tasks.

Response Parameters

None

Example Requests

Modifying dumping task quotas.

```
PUT https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks
{
  "sink_max_tasks" : 9
}
```

Example Response

None

SDK Sample Code

The SDK sample code is as follows.

Java

Modifying dumping task quotas.

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class UpdateSinkTaskQuotaSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateSinkTaskQuotaRequest request = new UpdateSinkTaskQuotaRequest();
        UpdateSinkTaskQuotaReq body = new UpdateSinkTaskQuotaReq();
        body.withSinkMaxTasks(9);
        request.withBody(body);
        try {
            UpdateSinkTaskQuotaResponse response = client.updateSinkTaskQuota(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

Modifying dumping task quotas.

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(KafkaRegion.value_of("<YOUR REGION>")) \  
.build()  
  
try:  
    request = UpdateSinkTaskQuotaRequest()  
    request.body = UpdateSinkTaskQuotaReq(  
        sink_max_tasks=9  
    )  
    response = client.update_sink_task_quota(request)  
    print(response)  
except exceptions.ClientRequestException as e:  
    print(e.status_code)  
    print(e.request_id)  
    print(e.error_code)  
    print(e.error_msg)
```

Go

Modifying dumping task quotas.

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := kafka.NewKafkaClient(  
        kafka.KafkaClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.UpdateSinkTaskQuotaRequest{}  
    request.Body = &model.UpdateSinkTaskQuotaReq{  
        SinkMaxTasks: int32(9),  
    }  
    response, err := client.UpdateSinkTaskQuota(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

Status Codes

Status Code	Description
204	Dumping task quotas are modified successfully.

Error Codes

See [Error Codes](#).

7.2.4.2 Creating an Order to Disable Dumping Connectors for an Instance

Function

This API is used to create an order to disable dumping connectors for an instance.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Disabling Smart Connect \(Pay-per-Use Instance\)](#).

Call Method

For details, see [How to Call an API](#).

URI

POST `/v2/{project_id}/kafka/instances/{instance_id}/delete-connector-order`

Table 7-65 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID . Minimum length: 1 Maximum length: 100
instance_id	Yes	String	Instance ID. Minimum length: 1 Maximum length: 100

Request Parameters

Table 7-66 Request body parameters

Parameter	Mandatory	Type	Description
instance_id	Yes	String	ID of the instance whose connector you want to disable, which is the same as that in the request path.
url	No	String	Page displayed after the order for disabling connector is submitted.

Response Parameters

Status code: 200

Table 7-67 Response body parameter

Parameter	Type	Description
order_id	String	Order ID generated by CBC.

Example Request

```
POST https://{endpoint}/v2/{project_id}/kafka/instances/{instance_id}/delete-connector-order
{
  "instance_id": "20c6b355-5d95-45ef-b476-e38bccce0d7",
  "url": "https://console.xxx.xxx.com/dms/?engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList"
}
```

Example Response

Status code: 200

An order created to disable dumping connectors for an instance

```
{
  "order_id": "CS2304180947HLABU"
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;
import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class CreateDeleteConnectorOrderSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDeleteConnectorOrderRequest request = new CreateDeleteConnectorOrderRequest();
        ConnectorOrderRequestBody body = new ConnectorOrderRequestBody();
        body.withUrl("https://console.xxx.xxx.com/dms/?engine=all&region=xxx&locale=xxx#/queue/manager/newKafkaList");
        body.withInstanceId("20c6b355-5d95-45ef-b476-e38bccce0d7");
        request.withBody(body);
        try {
            CreateDeleteConnectorOrderResponse response = client.createDeleteConnectorOrder(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \

client = KafkaClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateDeleteConnectorOrderRequest()
    request.body = ConnectorOrderRequestBody(
        url="https://console.xxx.xxx.com/dms/?engine=all&region=xxx&locale=xxx#/queue/manager/
newKafkaList",
        instance_id="20c6b355-5d95-45ef-b476-e38bcccc0d7"
    )
    response = client.create_delete_connector_order(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDeleteConnectorOrderRequest{}
    urlConnectorOrderRequestBody := "https://console.xxx.xxx.com/dms/?engine=all&region=xxx&locale=xxx#/
queue/manager/newKafkaList"
    request.Body = &model.ConnectorOrderRequestBody{
        Url: &urlConnectorOrderRequestBody,
        InstanceId: "20c6b355-5d95-45ef-b476-e38bcccc0d7",
    }
    response, err := client.CreateDeleteConnectorOrder(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Status Code

Status Code	Description
200	An order to disable dumping connectors for an instance is created.

Error Codes

See [Error Codes](#).

7.2.4.3 Creating a Dumping Task

Function

This API is used to create a dumping task.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Enabling Smart Connect \(Pay-per-Use Instance\)](#).

Call Method

For details, see [How to Call an API](#).

URI

POST `/v2/{project_id}/connectors/{connector_id}/sink-tasks`

Table 7-68 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
connector_id	Yes	String	Instance dump ID. For details, see Querying Instance IDs .

Request Parameters

Table 7-69 Request body parameters

Parameter	Mandatory	Type	Description
source_type	Yes	String	Source data type. Currently, only BLOB is supported.

Parameter	Mandatory	Type	Description
task_name	Yes	String	Name of a dumping task.
destination_type	Yes	String	Storage type. Currently, only OBS is supported.
obs_destination_descriptor	Yes	ObsDestinationDescriptor object	Description of the dump.

Table 7-70 ObsDestinationDescriptor

Parameter	Mandatory	Type	Description
topics	Yes	String	Names of the topics to be dumped. Multiple topics can be separated by commas (,). Regular expressions are supported.
topics_regex	No	String	Regular expression of the topics to be dumped. Set either the topics_regex parameter or the topics parameter. Do not set both of them. Do not set it to ".*".
consumer_strategy	Yes	String	Offset. <ul style="list-style-type: none"> • latest: Messages are consumed from the end of the topic. • earliest: Messages are consumed from the start of the topic. The default value is latest .
destination_file_type	Yes	String	Dump file format. Currently, only TXT files are supported.
access_key	Yes	String	Access key ID (AK).
secret_key	Yes	String	Secret access key (SK).
obs_bucket_name	Yes	String	Name of the OBS bucket used to store the data.

Parameter	Mandatory	Type	Description
obs_path	No	String	<p>OBS path. This parameter can be left blank by default.</p> <p>The value can contain a maximum of 64 characters, including letters, digits, underscores (_), hyphens (-), and slashes (/).</p> <p>This parameter is left empty by default.</p>
partition_for mat	Yes	String	<p>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).</p> <ul style="list-style-type: none"> • 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, 2017/11/10/14/49 means that the directory structure is 2017 > 11 > 10 > 14 > 49, where 2017 is the outermost folder. <p>NOTE After the data is dumped successfully, the storage directory structure is obs_bucket_path/file_prefix/partition_format. The default time zone is GMT+08:00.</p>

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. Value range: <ul style="list-style-type: none"> • Comma (,) • Semicolon (;) • Vertical bar () • Newline (\n) • NULL Default value: newline (\n).
deliver_time_interval	Yes	Integer	No package files will be generated if there is no data within a time segment. Value range: 30 to 900 Unit: second. NOTE This parameter is mandatory if streaming data is dumped to OBS.

Response Parameters

Status code: 200

Table 7-71 Response body parameter

Parameter	Type	Description
task_id	String	Task ID.

Example Request

Creating a dumping task to dump data in topic-test to OBS

POST https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks

```
{
  "source_type": "BLOB",
  "task_name": "obsTransfer-1122976956",
  "destination_type": "OBS",
  "obs_destination_descriptor": {
    "consumer_strategy": "earliest",
    "destination_file_type": "TEXT",
    "access_key": "XXXXXXXXXXXXXXXXXXXX",
    "secret_key": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "obs_bucket_name": "6666",
    "obs_path": "obsTransfer-1122976956",
  }
}
```



```
.withPartitionFormat(ObsDestinationDescriptor.PartitionFormatEnum.fromValue("yyyy/MM/dd/HH/mm"))
.withRecordDelimiter("")
.withDeliverTimeInterval(300);
body.withObsDestinationDescriptor(obsDestinationDescriptorbody);
body.withDestinationType(CreateSinkTaskReq.DestinationTypeEnum.fromValue("OBS"));
body.withTaskName("obsTransfer-1122976956");
body.withSourceType(CreateSinkTaskReq.SourceTypeEnum.fromValue("BLOB"));
request.withBody(body);
try {
    CreateSinkTaskResponse response = client.createSinkTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

Creating a dumping task to dump data in topic-test to OBS

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateSinkTaskRequest()
        obsDestinationDescriptorbody = ObsDestinationDescriptor(
            topics="topic-test",
            consumer_strategy="earliest",
            destination_file_type="TEXT",
            access_key="XXXXXXXXXXXXXXXXXXXX",
            secret_key="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
            obs_bucket_name="6666",
            obs_path="obsTransfer-1122976956",
            partition_format="yyyy/MM/dd/HH/mm",
            record_delimiter="",
            deliver_time_interval=300
        )
        request.body = CreateSinkTaskReq(
            obs_destination_descriptor=obsDestinationDescriptorbody,
```

```

        destination_type="OBS",
        task_name="obsTransfer-1122976956",
        source_type="BLOB"
    )
    response = client.create_sink_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

Creating a dumping task to dump data in topic-test to OBS

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateSinkTaskRequest{
        obsPathObsDestinationDescriptor:= "obsTransfer-1122976956"
        recordDelimiterObsDestinationDescriptor:= ""
        obsDestinationDescriptorbody := &model.ObsDestinationDescriptor{
            Topics: "topic-test",
            ConsumerStrategy: model.GetObsDestinationDescriptorConsumerStrategyEnum().EARLIEST,
            DestinationFileType: model.GetObsDestinationDescriptorDestinationFileTypeEnum().TEXT,
            AccessKey: "XXXXXXXXXXXXXXXXXXXX",
            SecretKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
            ObsBucketName: "6666",
            ObsPath: &obsPathObsDestinationDescriptor,
            PartitionFormat: model.GetObsDestinationDescriptorPartitionFormatEnum().YYYY_MM_DD_HH_MM,
            RecordDelimiter: &recordDelimiterObsDestinationDescriptor,
            DeliverTimeInterval: int32(300),
        }
    }
    request.Body = &model.CreateSinkTaskReq{
        ObsDestinationDescriptor: obsDestinationDescriptorbody,
        DestinationType: model.GetCreateSinkTaskReqDestinationTypeEnum().OBS,
        TaskName: "obsTransfer-1122976956",
        SourceType: model.GetCreateSinkTaskReqSourceTypeEnum().BLOB,
    }
    response, err := client.CreateSinkTask(request)
    if err == nil {

```

```

    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}

```

Status Code

Status Code	Description
200	A dumping task is created successfully.

Error Codes

See [Error Codes](#).

7.2.4.4 Listing Dumping Tasks

Function

This API is used to list dumping tasks.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Querying Smart Connect Tasks](#).

Call Method

For details, see [How to Call an API](#).

URI

GET `/v2/{project_id}/connectors/{connector_id}/sink-tasks`

Table 7-72 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
connector_id	Yes	String	Instance dump ID. For details, see Querying Instance IDs .

Request Parameters

None

Response Parameters

Status code: 200

Table 7-73 Response body parameters

Parameter	Type	Description
tasks	Array of tasks objects	List of dumping tasks.
total_number	Integer	Total number of dumping tasks.
max_tasks	Integer	Maximum number of tasks.
quota_tasks	Integer	Task quotas.

Table 7-74 tasks

Parameter	Type	Description
task_id	String	Task ID.
task_name	String	Name of a dumping task.
destination_type	String	Type of the dumping task.
create_time	Long	Time when the dumping task is created.
status	String	Dumping task status.
topics	String	Topic list or topic regular expression of the dumping task.

Example Request

Listing dumping tasks

```
GET https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks
```

Example Response

Status code: 200

Successfully listing dumping tasks

```
{
  "tasks": [ {
    "task_id": "2e148bed-3038-4617-8ade-b52e84a33eeb",
    "task_name": "obsTransfer-1122976956",
    "destination_type": "OBS",
    "create_time": 1592309487621,
    "status": "RUNNING",
    "topics": "topic-test"
  } ],
}
```

```
"total_number" : 1,  
"max_tasks" : 9,  
"quota_tasks" : 10  
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;  
import com.huaweicloud.sdk.kafka.v2.*;  
import com.huaweicloud.sdk.kafka.v2.model.*;  
  
public class ListSinkTasksSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        KafkaClient client = KafkaClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListSinkTasksRequest request = new ListSinkTasksRequest();  
        try {  
            ListSinkTasksResponse response = client.listSinkTasks(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion  
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListSinkTasksRequest()
        response = client.list_sink_tasks(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := kafka.NewKafkaClient(
        kafka.KafkaClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListSinkTasksRequest{}
    response, err := client.ListSinkTasks(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

Status Code

Status Code	Description
200	Dumping tasks are listed successfully.

Error Codes

See [Error Codes](#).

7.2.4.5 Querying a Dumping Task

Function

This API is used to query a dumping task.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Querying Smart Connect Tasks](#).

Call Method

For details, see [How to Call an API](#).

URI

GET `/v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}`

Table 7-75 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .
connector_id	Yes	String	Instance dump ID. For details, see Querying Instance IDs .
task_id	Yes	String	Dumping task ID.

Table 7-76 Query parameters

Parameter	Mandatory	Type	Description
topic-info	No	String	Whether topic information is contained. The default value is false .

Request Parameters

None

Response Parameters

Status code: 200

Table 7-77 Response body parameters

Parameter	Type	Description
task_name	String	Name of a dumping task.
destination_type	String	Type of the dumping task.
create_time	Long	Time when the dumping task is created.
status	String	Dumping task status.
topics	String	Topic list or topic regular expression of the dumping task.
obs_destination_descriptor	obs_destination_descriptor object	Description of the dump.
topics_info	Array of topics_info objects	Topic information.

Table 7-78 obs_destination_descriptor

Parameter	Type	Description
consumer_strategy	String	Message consumption policy: <ul style="list-style-type: none"> • latest: Messages are consumed from the end of the topic. • earliest: Messages are consumed from the start of the topic. The default value is latest .
destination_file_type	String	Dump file format. Only text is supported.
obs_bucket_name	String	Name of the OBS bucket used to store the data.
obs_path	String	OBS path.

Parameter	Type	Description
partition_format	String	<p>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 dumping task was created).</p> <ul style="list-style-type: none"> • 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, 2017/11/10/14/49 means that the directory structure is 2017 > 11 > 10 > 14 > 49, where 2017 is the outermost folder. <p>NOTE After the data is dumped successfully, the storage directory structure is obs_bucket_path/file_prefix/partition_format. The default time zone is GMT +08:00.</p>
record_delimiter	String	<p>Delimiter for the dump file, which is used to separate the user data that is written into the dump file.</p> <p>Value range:</p> <ul style="list-style-type: none"> • Comma (,) • Semicolon (;) • Vertical bar () • Newline (\n) • NULL <p>Default value: newline (\n).</p>
deliver_time_interval	Integer	<p>No package files will be generated if there is no data within a time segment.</p> <p>Value range: 30-900. Default value: 300. Unit: second.</p> <p>NOTE This parameter is mandatory if streaming data is dumped to OBS.</p>
obs_part_size	Long	<p>Size (in bytes) of each file to be uploaded.</p> <p>Default value: 5242880.</p>

Table 7-79 topics_info

Parameter	Type	Description
topic	String	Topic name.
partitions	Array of partitions objects	Partition list.

Table 7-80 partitions

Parameter	Type	Description
partition_id	String	Partition ID.
status	String	Running status.
last_transfer_offset	String	Dumped message offset.
log_end_offset	String	Message offset.
lag	String	Number of stacked messages.

Example Request

Querying specified dumping task details

```
GET https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}?topic-info=true
```

Example Response

Status code: 200

Successfully querying a dumping task

```
{
  "task_name": "obsTransfer-56997523",
  "destination_type": "OBS",
  "create_time": 1628126621283,
  "status": "RUNNING",
  "topics": "topic-sdk-no-delete",
  "obs_destination_descriptor": {
    "consumer_strategy": "earliest",
    "destination_file_type": "TEXT",
    "obs_bucket_name": "testobs",
    "obs_path": "obsTransfer-56997523",
    "partition_format": "yyyy/MM/dd/HH/mm",
    "record_delimiter": "",
    "deliver_time_interval": 300,
    "obs_part_size": 5242880
  },
  "topics_info": [ {
    "topic": "topic-sdk-no-delete",
    "partitions": [ {
      "partition_id": "2",
```

```
"status" : "RUNNING",
"last_transfer_offset" : "3",
"log_end_offset" : "3",
"lag" : "0"
}, {
"partition_id" : "1",
"status" : "RUNNING",
"last_transfer_offset" : "3",
"log_end_offset" : "3",
"lag" : "0"
}, {
"partition_id" : "0",
"status" : "RUNNING",
"last_transfer_offset" : "3",
"log_end_offset" : "3",
"lag" : "0"
}
}
}
}
```

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class ShowSinkTaskDetailSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        KafkaClient client = KafkaClient.newBuilder()
            .withCredential(auth)
            .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowSinkTaskDetailRequest request = new ShowSinkTaskDetailRequest();
        request.withTopicInfo(ShowSinkTaskDetailRequest.TopicInfoEnum.fromValue("<topic-info>"));
        try {
            ShowSinkTaskDetailResponse response = client.showSinkTaskDetail(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowSinkTaskDetailRequest()
        request.topic_info = "<topic-info>"
        response = client.show_sink_task_detail(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()
```

```

client := kafka.NewKafkaClient(
    kafka.KafkaClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowSinkTaskDetailRequest{}
topicInfoRequest := model.GetShowSinkTaskDetailRequestTopicInfoEnum().<TOPIC_INFO>
request.TopicInfo = &topicInfoRequest
response, err := client.ShowSinkTaskDetail(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

Status Code

Status Code	Description
200	The dumping task is queried successfully.

Error Codes

See [Error Codes](#).

7.2.4.6 Deleting a Dumping Task

Function

This API is used to delete a dumping task.

NOTE

This API is out-of-date and may not be maintained in the future. Please use the API described in [Deleting a Smart Connect Task](#).

Call Method

For details, see [How to Call an API](#).

URI

DELETE /v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}

Table 7-81 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
connector_id	Yes	String	Instance dump ID. For details, see Querying Instance IDs .
task_id	Yes	String	Dumping task ID.

Request Parameters

None

Response Parameters

None

Example Request

Deleting a specified dumping task

```
DELETE https://{endpoint}/v2/{project_id}/connectors/{connector_id}/sink-tasks/{task_id}
```

Example Response

None

SDK Sample Code

The SDK sample code is as follows.

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.kafka.v2.region.KafkaRegion;
import com.huaweicloud.sdk.kafka.v2.*;
import com.huaweicloud.sdk.kafka.v2.model.*;

public class DeleteSinkTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);
```

```
KafkaClient client = KafkaClient.newBuilder()
    .withCredential(auth)
    .withRegion(KafkaRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteSinkTaskRequest request = new DeleteSinkTaskRequest();
try {
    DeleteSinkTaskResponse response = client.deleteSinkTask(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddkafka.v2.region.kafka_region import KafkaRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddkafka.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = KafkaClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(KafkaRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteSinkTaskRequest()
        response = client.delete_sink_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    kafka "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/kafka/v2/region"
)
```

```
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := kafka.NewKafkaClient(  
        kafka.KafkaClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DeleteSinkTaskRequest{}  
    response, err := client.DeleteSinkTask(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

Status Code

Status Code	Description
204	The dumping task is deleted successfully.

Error Codes

See [Error Codes](#).

8 Appendix

8.1 Status Code

[Table 8-1](#) lists status codes.

Table 8-1 Status codes

Status Code	Name	Description
100	Continue	The server has received the initial part of the request and the client should continue to send the remaining part.
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	Request sent successfully.
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 request has been fulfilled.
204	NoContent	The server has successfully processed the request, but is not returning any response body. The status code is returned in response to an HTTP OPTIONS request.

Status Code	Name	Description
205	Reset Content	The server has fulfilled the request, but the requester is required to reset the content.
206	Partial Content	The server has successfully processed a part of the GET request.
300	Multiple Choices	There are multiple options for the 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 have been permanently moved to the given URI indicated in this response.
302	Found	The requested resource was temporarily moved.
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. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource is available only through a proxy.
306	Unused	This HTTP status code is no longer used.
400	BadRequest	Invalid request. The client should modify the request instead of re-initiating it.
401	Unauthorized	The authorization 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 cannot be found. The client should modify the request instead of re-initiating it.

Status Code	Name	Description
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 cannot fulfill the request based on 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 when waiting for the request. The client may re-initiate the request without any modification at any time.
409	Conflict	The request cannot be processed due to a conflict, such as 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.
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 server refuses to process a request because the request is too large. The server may close the connection to prevent the client from continuing the request. If the server cannot process the request temporarily, the response will contain a Retry-After 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.

Status Code	Name	Description
422	UnprocessableEntity	The request is well-formed but is unable to be processed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given amount of time, or the server has received more requests than it is able to process within a given amount of time. In this case, the client should re-initiate requests after the time specified in the Retry-After header of the response expires.
500	InternalServerError	The server is able to receive the request but it could not understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server 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 request cannot be fulfilled within a given time. The response will reach the client only if the request carries the timeout parameter.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

8.2 Error Codes

If an error code starting with APIGW is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400002	The project ID format is invalid.	Invalid project ID.	Check the project ID format.
400	DMS.00400004	The request body is empty.	The request body is empty.	Check the request body.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400005	The message body is not in JSON format or contains invalid characters.	Check the project ID format.	Check the message body format.
400	DMS.00400007	Unsupported type.	Unsupported type.	Check the type.
400	DMS.00400008	Unsupported version.	Unsupported version.	Check the version.
400	DMS.00400009	Invalid product_id.	Invalid product_id in the request.	Check the product_id parameter.
400	DMS.00400010	Invalid instance name. The name must be 4 to 64 characters long. Only letters, digits, underscores (_), and hyphens (-) are allowed.	Invalid instance name. The name must be 4 to 64 characters long. Only letters, digits, underscores (_), and hyphens (-) are allowed.	Check the instance name.
400	DMS.00400011	The instance description can contain a maximum of 1024 characters.	The instance description can contain a maximum of 1024 characters.	Check the instance description.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.0040001 2	The password does not meet the complexity requirements. An instance password must: Be a string consisting of 8 to 32 characters. Contain at least two of the following character types: Lowercase letters Uppercase letters Digits Special characters `~!@#\$%^&*()-_+=\ [{ }:; ',<.>/?	The password does not meet the complexity requirements. A password: - Can contain 8 to 32 characters. - Must contain at least three of the following character types: letters, digits, and special characters `~!@#\$%^&*()-_+=\ [{ }:; ',<.>/? - Cannot be a weak password.	Check whether the password meets the requirements.
400	DMS.0040001 3	vpc_id in the request is empty.	Request parameter vpc_id is empty.	Check the vpc_id parameter.
400	DMS.0040001 4	security_group_id in the request is empty.	Request parameter security_group_id is empty.	Check the security_group_id parameter.
400	DMS.0040001 5	Invalid username. A username must be 4 to 64 characters long and consist of only letters, digits, and hyphens (-).	Invalid username. A username must be 4 to 64 characters long and consist of only letters, digits, and hyphens (-).	Check the username.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400016	subnet_id in the request is empty.	Request parameter subnet_id is empty.	Check the subnet_id parameter.
400	DMS.00400017	This DMS instance job task is still running.	A background task associated with this instance is running.	Try again later.
400	DMS.00400018	This subnet must exist in the VPC.	The subnet must exist in the VPC.	Check the subnet.
400	DMS.00400019	The password does not meet the complexity requirements.	The password does not meet the complexity requirements.	Check whether the password meets the requirements.
400	DMS.00400020	DHCP must be enabled for this subnet.	DHCP must be enabled for the subnet.	Check the DHCP status.
400	DMS.00400021	The isAutoRenew parameter in the request must be either 0 or 1.	Invalid isAutoRenew in the request.	Check the isAutoRenew parameter.
400	DMS.00400022	Engine does not match the product id.	The engine and product ID parameters do not match.	Check the engine parameter.
400	DMS.00400026	This operation is not allowed due to the instance status.	This operation is not allowed when the instance is in the current state.	Check the instance status.
400	DMS.00400028	Query advanced product, specCode not exists.	The specCode does not exist during the advanced feature query.	Check the origin_spec_code parameter.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400029	Query advanced product failed, can not find product for request.	The specCode does not exist during the advanced feature query.	Check the origin_spec_code parameter.
400	DMS.00400030	Invalid DMS instance id. The id must be a uuid.	Invalid instance ID.	Check the id parameter.
400	DMS.00400035	DMS instance quota of the tenant is insufficient.	Insufficient instance quota.	Apply for a higher quota.
400	DMS.00400037	The instanceParams parameter in the request contains invalid characters or is not in JSON format.	Request parameter instanceParams is not in JSON format or contains invalid characters.	Check the request parameter.
400	DMS.00400038	The periodNum parameter in the request must be an integer.	The periodNum parameter in the request must be an integer.	Check the periodNum parameter.
400	DMS.00400039	The quota limit has been reached.	The quota limit has been reached.	Apply for a higher quota.
400	DMS.00400042	The AZ does not exist.	The AZ does not exist.	Check the AZ.
400	DMS.00400045	The instance is not frozen and cannot be unfrozen.	The instance cannot be unfrozen because it is not frozen.	Check the instance status.
400	DMS.00400046	This security group does not exist.	The security group does not exist.	Check the security group.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400047	The periodType parameter in the request must be either 2 or 3.	Invalid periodType in the request.	Check the periodType parameter.
400	DMS.00400048	Invalid security group rules. Ensure that rules with the protocol being ANY are configured for both the inbound and outbound directions.	The security group must have both outbound and inbound rules with protocols set to ANY.	Check the security group rules.
400	DMS.00400049	The availability zone does not support ipv6.	The AZ does not support IPv6.	Select another AZ.
400	DMS.00400051	not found the new setup version tar to upgrade instance.	The package for upgrading the instance to the target version is not found.	Select another target version.
400	DMS.00400052	only the instance at running status can upgrade.	Only running instances can be upgraded.	Try again later.
400	DMS.00400053	the upgrade instance version equals to current version.	The target version is the same as the current version.	Select another target version.
400	DMS.00400055	Resource sold out.	Resources, such as ECS and volume resources, are insufficient.	Try again later.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400060	This instance name already exists.	The instance name already exists.	Check the instance name.
400	DMS.00400061	Invalid instance ID format.	Invalid instance ID.	Check the instance ID.
400	DMS.00400062	Invalid request parameter.	Invalid request parameters.	Check the request parameters.
400	DMS.00400063	Invalid configuration parameter {0}.	Invalid configuration parameter {0}.	Check the parameter.
400	DMS.00400064	The action parameter in the request must be delete or restart.	The action parameter in the request must be delete or restart.	Check the action parameter.
400	DMS.00400065	The instances parameter in the request is empty.	The instances parameter in the request is empty.	Check the instances parameter.
400	DMS.00400066	Invalid configuration parameter {0}.	Invalid configuration parameter {0}.	Check the parameter.
400	DMS.00400067	The available_zones parameter in the request must be an array that contains only one AZ ID.	Request parameter available_zones must be an array that contains only one AZ ID.	Check the available_zones parameter.
400	DMS.00400068	The VPC does not exist.	The VPC does not exist.	Check the VPC.
400	DMS.00400070	Invalid task ID format.	Invalid task ID.	Check the task ID.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400077	Insufficient IPs in the selected subnet.	Too few IP addresses in the selected subnet.	Select another subnet with sufficient IP addresses.
400	DMS.00400081	Duplicate instance name.	The instance name already exists.	Check the instance name.
400	DMS.00400082	Instance id is repeated.	The instance ID already exists.	Check the instance ID.
400	DMS.00400085	The message body contains invalid characters or is not in JSON format. The error key is <key>.	The message body is not in JSON format or contains invalid characters.	Check the message body.
400	DMS.00400099	The following instances in the Creating, Starting, Stopping, or Restarting state cannot be deleted.	Instances ({} in the Creating, Starting, Stopping, or Restarting state cannot be deleted.	Check the instance status.
400	DMS.00400100	The instances array can contain a maximum of 50 instance IDs.	The instances array can contain a maximum of 50 instance IDs.	Check the instance quantity.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.0040010 1	The name of a Kafka topic must be 4 to 64 characters long and start with a letter. Only letters, digits, underscores (_), and hyphens (-) are allowed.	The name of a topic in a Kafka instance must be 4 to 64 characters long and start with a letter. Only letters, digits, underscores (_), and hyphens (-) are allowed.	Check the topic name.
400	DMS.0040010 2	The number of partitions created for a Kafka topic must be within the range of 1-200.	The number of partitions created for a topic in a Kafka instance must be within the range of 1-200.	Check the number of partitions of the topic.
400	DMS.0040010 3	The number of replicas created for a Kafka topic must be within the range of 1-20.	The number of replicas created for a topic in a Kafka instance must be within the range of 1-20.	Check the number of replicas of the topic.
400	DMS.0040010 5	The message retention period of a Kafka topic must be within the range of 1-168.	The aging time of a topic in a Kafka instance must be within the range of 1-168.	Check the aging time of the topic.
400	DMS.0040010 6	Invalid maintenance time window.	Invalid maintenance time window.	Check the maintenance time window parameter.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400107	The instance exists for unpaid scale up orders. Please process non payment orders first.	A specification modification order for the instance is pending.	Process the order first.
400	DMS.00400108	The Instance exists for processing scale up order. Please try again later.	A specification modification order for the instance is being processed.	Try again later.
400	DMS.00400124	The maximum number of disk expansion times has been reached.	The maximum number of disk expansion times has been reached.	Check the maximum number of disk expansion times.
400	DMS.00400125	Invalid SPEC_CODE.	Invalid SPEC_CODE.	Check SPEC_CODE.
400	DMS.00400126	Invalid period time.	Invalid time period for yearly/monthly billing.	Check the time period for yearly/monthly billing.
400	DMS.00400127	Instance not support to change retention_policy.	The instance does not support retention policy changes.	Contact technical support.
400	DMS.00400128	Invalid public access parameters.	Invalid public access parameters.	Check the public access parameters.
400	DMS.00400129	Current instance version is less than required.	The instance version does not support this operation.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400133	Sink task quota for connector invalid.	Invalid connector task quota.	Contact technical support.
400	DMS.00400134	There is another order need to pay first.	An unpaid order exists.	Pay for the order first.
400	DMS.00400135	Not support disk encrypted.	Disk encryption is not supported.	Do not enable disk encryption.
400	DMS.00400136	Disk encrypted key is null.	The disk encryption key is empty.	Check the disk encryption key.
400	DMS.00400137	Disk encrypted key state is not enabled.	The disk encryption key is not enabled.	Enable the disk encryption key.
400	DMS.00400142	Timestamp is invalid.	Timestamp is invalid.	Enter a correct timestamp.
400	DMS.00400500	Invalid disk space.	Invalid disk space.	Check the disk space.
400	DMS.00400800	Invalid request parameter. Check the request parameter.	Invalid request parameter.	Check the request parameters.
400	DMS.00400861	Replication factor larger than available brokers.	The number of replicas in the topic to be created is greater than the number of available brokers.	Contact technical support.
400	DMS.00400867	Failed to create the Smart Connect task.	Failed to create the Smart Connect task.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00400868	Failed to stop the Smart Connect task.	Failed to stop the Smart Connect task.	Try again later.
400	DMS.00400869	Failed to start the Smart Connect task.	Failed to start the Smart Connect task.	Try again later.
400	DMS.00400870	Failed to verify the Smart Connect task.	Failed to verify the Smart Connect task.	Try again later.
400	DMS.00400872	Failed to restart the Smart Connect task.	Failed to restart the Smart Connect task.	Try again later.
400	DMS.00400873	Failed to modify the Smart Connect task.	Failed to modify the Smart Connect task.	Contact technical support.
400	DMS.00400874	The topic has been used in another Smart Connect task.	The topic has been used in another Smart Connect task.	Check the topic and try again.
400	DMS.00400875	Inconsistent source and target Redis instance types in the Smart Connect task.	Inconsistent source and target Redis instance types in the Smart Connect task.	Change the source and target Redis instance types and try again.
400	DMS.00400876	The topic does not exist.	The topic does not exist.	Check the topic and try again.
400	DMS.00400970	RabbitMQ plugin is not exist	Invalid plugin name	Check the plugin list.
400	DMS.00400971	The instance ssl is off.	The instance ssl is off.	View the instance details and check whether SSL is enabled.
400	DMS.00400975	Failed to query topics.	Failed to query topics.	Check whether the topic exists.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.00404033	Does not support extend rabbitMQ disk space.	The RabbitMQ disk space cannot be expanded.	Scale out the RabbitMQ cluster.
400	DMS.00500033	Failed to access EPS to update the project	Failed to access EPS to update the project	Contact technical support.
400	DMS.00500960	Invalid user AK/SK.	Invalid user AK/SK.	Correct the user AK/SK and try again.
400	DMS.00500986	Your account has been restricted.	Your account is suspended.	Contact the billing center.
400	DMS.00500987	Balance is not enough	Insufficient balance.	Top up your account and try again later.
400	DMS.10240002	The number of queried queues exceeds the upper limit.	The maximum number of queried queues has been reached.	Check the queue quantity.
400	DMS.10240004	The tag name is invalid.	Invalid tag name.	Check the tag name.
400	DMS.10240005	The project ID format is invalid.	Invalid project ID.	Check the project ID format.
400	DMS.10240007	The name contains invalid characters.	The name contains invalid characters.	Check the name.
400	DMS.10240009	The message body is not in JSON format or contains invalid characters.	The message body is not in JSON format or contains invalid characters.	Check the message body.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.10240010	The description contains invalid characters.	The description contains invalid characters.	Check the description.
400	DMS.10240011	The name length must be 1 to 64 characters.	The name can contain 1 to 64 characters.	Check the name length.
400	DMS.10240012	The name length must be 1 to 32 characters.	The name can contain 1 to 32 characters.	Check the name length.
400	DMS.10240013	The description length must not exceed 160 characters.	The description can contain a maximum of 160 characters.	Check the description length.
400	DMS.10240014	The number of consumable messages exceeds the maximum limit.	The number of consumable messages is not within the allowed range.	Check the number of consumable messages.
400	DMS.10240015	The queue ID format is invalid.	Invalid queue ID.	Check the queue ID.
400	DMS.10240016	The group ID format is invalid.	Invalid group ID.	Check the group ID.
400	DMS.10240017	The queue already exists.	The queue already exists.	Check whether the queue exists.
400	DMS.10240018	The consumer group already exists.	The consumer group already exists.	Check whether the consumer group exists.
400	DMS.10240019	The number of consumer groups exceeds the upper limit.	The number of consumer groups exceeds the upper limit.	Check the number of consumer groups.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.10240020	The quota is insufficient.	Insufficient quota.	Check the quota.
400	DMS.10240021	The value of time_wait is not within the value range of 1-60.	The value of time_wait is not within the range of 1-60.	Check the value of time_wait.
400	DMS.10240022	The value of max Consume Count must be within the range of 1-100.	The value of max Consume Count must be within the range of 1-100.	Check the value of max Consume Count.
400	DMS.10240027	The value of retention_hours must be an integer in the range of 1-72.	The value of retention_hours must be an integer in the range of 1-72.	Check the value of retention_hours.
400	DMS.10240028	Non-kafka queues do not support retention_hours.	Non-kafka queues do not support retention_hours.	Check whether the queue is a Kafka queue. If not, do not set retention_hours.
400	DMS.10240032	The queue is being created.	The queue is being created.	Check whether the queue is being created.
400	DMS.10240035	The tag key is empty or too long.	The tag key of the queue is empty or too long.	Check the tag key of the queue.
400	DMS.10240036	The tag key contains invalid characters.	The tag key of the queue contains invalid characters.	Check the tag key of the queue.
400	DMS.10240038	The tag value is too long.	The tag value is too long.	Check the tag value of the queue.
400	DMS.10240039	The tag value contains invalid characters.	The tag value contains invalid characters.	Check the tag value of the queue.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.10240040	You can only create or delete tags.	You can only create or delete tags.	Check whether the operation meets the requirements.
400	DMS.10240041	You can only filter or count tags.	You can only filter or count tags.	Check whether the operation meets the requirements.
400	DMS.10240042	The number of records on each page for pagination query exceeds the upper limit.	The number of records on each page for pagination query exceeds the upper limit.	Check the page size.
400	DMS.10240043	The number of skipped records for pagination query exceeds the upper limit.	The offset for pagination query exceeds the upper limit.	Check the paging offset.
400	DMS.10240044	A maximum of 10 tags can be created.	A maximum of 10 tags can be created.	Check the tag quantity.
400	DMS.10240045	The tag key has been used.	The tag key has been used.	Check whether the tag key has been used.
400	DMS.10540001	The message body contains invalid fields.	The message body contains invalid fields.	Check the message body.
400	DMS.10540003	Message ack status must be either 'success' or 'fail'. It should not be '{status}'.	Message ack status must be either success or fail. It should not be {status}.	Check whether the status meets the requirements.
400	DMS.10540004	Request error	Request error. The queue or group name does not match the handler.	Check whether the queue or group name matches the handler.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.10540010	The request format is incorrect	The request format is incorrect. {Error description}	Check the request format.
400	DMS.10540011	The message size is {message size}, larger than the size limit {max allowed size}.	The message size is {message size}, larger than the size limit {max allowed size}.	Check the request body size.
400	DMS.10540012	The message body is not in JSON format or contains invalid characters.	The message body is not in JSON format or contains invalid characters.	Check the message body format.
400	DMS.10540014	The URL contains invalid parameters.	The URL contains invalid parameters.	Check the URL parameters.
400	DMS.10540202	The request format is incorrect	The request format is incorrect. {Error description}	Check the request format.
400	DMS.10542204	Failed to consume messages due to {desc}.	Failed to consume messages. {Error description}	Check the error information and rectify the fault accordingly.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.10542205	Failed to obtain the consumption instance because the handler does not exist. This may be because the consumer instance is released 1 minute after the message is consumed. As a result, the consumer instance fails to be obtained from the handler.	Failed to obtain the consumption instance because the handler does not exist. This may be because the consumer instance is released 1 minute after the message is consumed.	Check the handler.
400	DMS.10542206	The value of ack_wait must be within the range of 15-300.	The value of ack_wait must be within the range of 15-300.	Check the value of ack_wait.
400	DMS.10542209	The handler does not exist because the handler fails to be parsed, the message consumption times out, or the message consumption is repeatedly acknowledged.	The handler does not exist because the handler fails to be parsed, the message consumption times out, or the message consumption is repeatedly acknowledged.	Check whether the handler or consumption acknowledgment times out.
400	DMS.10542214	The request format is incorrect	The request format is incorrect. {Error description}	Check the request format.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.111400860	Instance partition is not enough.Total partition is over the partition limitation.	Instance partition is not enough. Total partition is over the partition limitation.	Check whether the partition quantity is exceeded.
400	DMS.50050004	The consumer group is offline.	The consumer group is offline.	Start the consumer instance in the consumer group.
401	DMS.10240101	Invalid token.	Invalid token.	Check whether the token is valid.
401	DMS.10240102	Expired token.	The token has expired.	Check whether the token has expired.
401	DMS.10240103	Missing token.	The token is missing.	Check whether the token is missing.
401	DMS.10240104	The project ID and token do not match.	The project ID and token do not match.	Check whether the project ID matches the token.
403	DMS.00403002	A tenant has the read-only permission and cannot perform operations on DMS.	You cannot perform operations on DMS because you only have read permissions.	Check the tenant permission.
403	DMS.00403003	This role does not have the permissions to perform this operation.	This role does not have the permissions to perform this operation.	Check the role permission.
403	DMS.00403007	Authorization denied.	Action does not allow to performed.	PlESE check your permissions.

Status Code	Error Codes	Error Message	Description	Solution
403	DMS.10240304	Change the quota of a queue or consumer group to a value smaller than the used quota.	The quota of a queue or consumer group cannot be smaller than the used amount.	Check the quota.
403	DMS.10240306	The tenant has been frozen. You cannot perform operations on DMS.	The tenant has been frozen. You cannot perform operations on DMS.	Check the tenant status.
403	DMS.10240307	The consumer group quota must be within the range of 1-10.	The consumer group quota must be within the range of 1-10.	Check whether the number of consumer groups exceeds the quota.
403	DMS.10240308	The queue quota must be within the range of 1-20.	The queue quota must be within the range of 1-20.	Check whether the number of queues exceeds the quota.
403	DMS.10240309	Access denied. You cannot perform operations on DMS.	Access denied. You cannot perform operations on DMS.	Check whether you have the permission required to perform this operation.
403	DMS.10240310	A tenant has the read-only permission and cannot perform operations on DMS.	The tenant has read-only permissions and cannot perform operations on DMS.	Check the tenant permission.
403	DMS.10240311	This role does not have the permissions to perform this operation.	This role does not have the permissions required to perform operations on DMS.	Check the role permission.

Status Code	Error Codes	Error Message	Description	Solution
403	DMS.1024031 2	The tenant is restricted and cannot perform operations on DMS.	The tenant is restricted and cannot perform operations on DMS.	Check the role permission.
404	DMS.0040400 1	The requested URL does not exist.	The requested URL does not exist.	Check the URL.
404	DMS.0040402 2	This instance does not exist.	The instance does not exist.	Check whether the instance exists.
404	DMS.0040402 4	Connector does not exist.	Connector does not exist.	Check the connector.
404	DMS.0040402 6	The dumping task does not exist.	The dumping task does not exist.	Check the dumping task.
404	DMS.0040402 7	Connector already exists.	Connector already exists.	Check the connector.
404	DMS.0040402 9	The dumping task quota has been reached.	The dumping task quota has been reached.	Check the dumping task quota.
404	DMS.1024040 1	The queue ID is incorrect or not found.	The queue ID is incorrect or is not found.	Check whether the queue ID exists and is correct.
404	DMS.1024040 5	The consumption group ID is incorrect or not found.	The consumption group ID is incorrect or is not found.	Check whether the consumer group ID exists and is correct.
404	DMS.1024040 6	The URL or endpoint does not exist.	The URL or endpoint does not exist.	Check whether the URL or endpoint exists and is correct.

Status Code	Error Codes	Error Message	Description	Solution
404	DMS.10240407	The request is too frequent. Flow control is being performed. Please try again later.	The request is sent too frequently and flow control is being performed. Please try again later.	Try again later.
404	DMS.10240426	No tag containing this key exists.	No tags containing this key exist.	Check the tag.
404	DMS.10540401	The queue name does not exist.	The queue name does not exist.	Check whether the queue name exists.
405	DMS.00405001	This request method is not allowed.	The request method is not allowed.	Check the request method.
408	DMS.111501024	Query timed out	Message query timeout	Please query later
500	DMS.00500000	Internal service error.	Internal service error.	Contact technical support.
500	DMS.00500006	Internal service error.	Internal service error.	Contact technical support.
500	DMS.00500007	Internal service error.	Internal service error.	Contact technical support.
500	DMS.005000024	Internal service error.	Internal service error.	Contact technical support.
500	DMS.005000025	Internal service error.	Internal service error.	Contact technical support.
500	DMS.005000041	Internal service error.	Internal service error.	Contact technical support.
500	DMS.005000052	Internal service error.	Failed to submit the instance upgrade job.	Contact technical support.
500	DMS.005000053	Internal service error.	The specified instance node is not found.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
500	DMS.00500054	Internal service error.	Failed to generate the password.	Contact technical support.
500	DMS.00500070	Internal service error.	Failed to configure the instance.	Contact technical support.
500	DMS.00500071	Internal service error.	Failed to create the instance backup policy.	Contact technical support.
500	DMS.00500094	Internal service error.	Internal service error.	Contact technical support.
500	DMS.00500106	Internal service error.	Internal service error.	Contact technical support.
500	DMS.00500990	Failed to update topics.	Failed to update topics.	Contact technical support.
500	DMS.00501000	Failed to create agency, may be you do not have the agency permission.	Failed to create agency	check whether the current user has the agency permission.
500	DMS.00501001	Failed to get agency roleId.	Failed to get agency roleId.	retry the request later
500	DMS.00501002	Failed to query agency roleId.	Failed to query the role ID based on the role name.	Check whether the role name in the request is correct.
500	DMS.00501003	Failed to grant role to agency.	Failed to grant role to agency.	Try again later, or contact technical support
500	DMS.00501010	The product specification does not exist.	The product specification does not exist.	Contact technical support.
500	DMS.00501011	Failed to query the product ID from CBC.	Failed to query the product ID from CBC.	Contact technical support.

Status Code	Error Codes	Error Message	Description	Solution
500	DMS.0050101 2	Smart Connect tasks exist.	Smart Connect tasks exist.	Delete all Smart Connect tasks.
500	DMS.1025000 2	Internal service error.	Internal service error.	Contact technical support.
500	DMS.1025000 3	Internal service error.	Internal service error.	Contact technical support.
500	DMS.1025000 4	Internal service error.	Internal service error.	Contact technical support.
500	DMS.1025000 5	Internal communication error.	Internal communication error.	Contact technical support.
500	DMS.1025000 6	Internal service error.	Internal service error.	Contact technical support.
500	DMS.1055003 5	tag_type must be either or or and.	tag_type must be either 'or' or 'and'.	Check tag_type.
501	DMS.1115010 26	Query reach maximum byte	The total number of bytes in the query exceeds the upper limit.	Shorten the time range to ensure that the number of queried bytes does not exceed the limit, or use other methods to query data.
503	DMS.1115010 25	Query Busy. Please try again later.	Message query busy	Please query later

8.3 Instance Status

Table 8-2 Instance status description

Status	Description
CREATING	The instance is being created.
RUNNING	The instance is running properly. Instances in this state can provide services.
ERROR	The instance is not running properly.

Status	Description
RESTARTING	The instance is being restarted.
STARTING	The status between Frozen and Running .
EXTENDING	The instance specifications are being changed.
EXTENDEDFAILED	The instance specifications failed to be changed.
FROZEN	The instance has been frozen due to insufficient account balance. You can unfreeze the instance by topping up your account in My Order .
FREEZING	The status between Running and Frozen .
UPGRADING	The instance is being upgraded.
ROLLINGBACK	The instance is being rolled back.

8.4 Obtaining a Project ID

Scenario

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

- [Obtaining a Project ID by Calling an API](#)
- [Obtaining a Project ID on the Console](#)

Obtaining a Project ID by Calling an API

You can obtain a project ID by calling the API used to [query projects based on specified criteria](#).

The API used to obtain a project ID is **GET https://{Endpoint}/v3/projects**, where *{Endpoint}* indicates the IAM endpoint. You can obtain the IAM endpoint from [Regions and Endpoints](#). For details on API calling authentication, see [Authentication](#).

The following is an example response. The value of **id** in the **projects** section is the project ID:

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxx-xxx-xxx",
      "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 on the Console

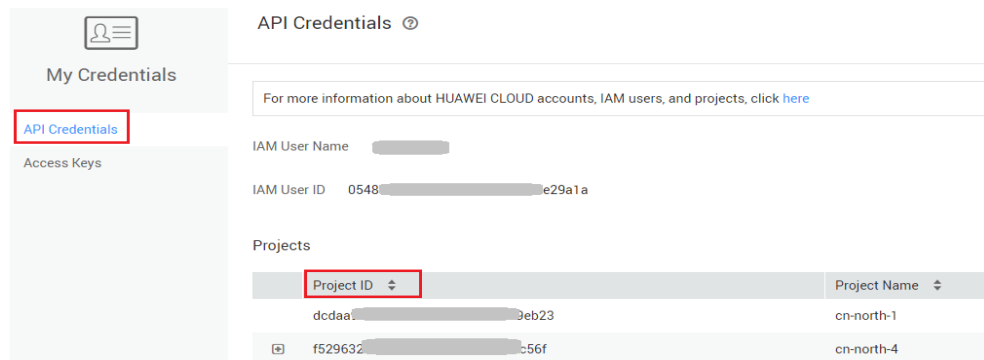
A project ID is required for some URLs when an API is called. You can obtain a project ID on the console.

The following procedure describes how to obtain a project ID:

- Step 1** Log in to the management console.
- Step 2** Hover the mouse pointer over the username in the upper right corner and choose **My Credentials** from the drop-down list.

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

Figure 8-1 Viewing a project ID



----End

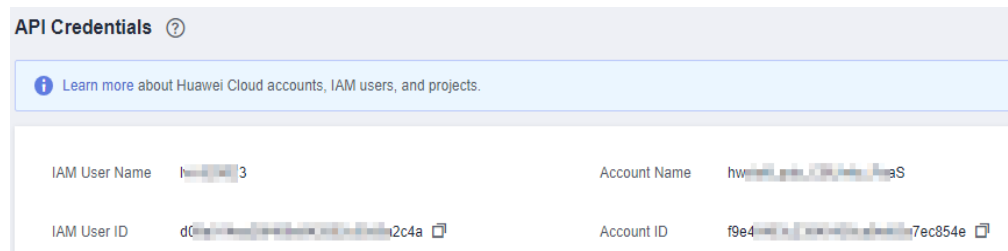
8.5 Obtaining the Account Name and Account ID

The account name and account ID are required for some URLs when an API is called. The following procedure describes how to obtain the domain and domain ID:

1. Log in to the console.
2. Hover the mouse pointer over the username in the upper right corner and choose **My Credentials** from the drop-down list.

View the account name and account ID.

Figure 8-2 Viewing the account name and account ID



A Change History

Released On	Description
2025-02-11	This issue incorporates the following change: <ul style="list-style-type: none"> Added the port_protocol parameter in Creating an instance.
2024-12-24	This release incorporates the following change: <ul style="list-style-type: none"> Added the APIs for Reassigning Replicas of a Topic for a Kafka Instance, Querying the Offset of a Consumer Group, Modifying All Consumer Groups, Querying a Specified Consumer Group, Deleting a Specified Consumer Group, Querying Topics of a Specified Consumer Group, Querying Consumers in a Specified Consumer Group, Deleting a Kafka Message, and Querying vCPUs of a Kafka Flavor.
2024-06-04	This issue incorporates the following change: <ul style="list-style-type: none"> Added APIs for Diagnosis management.
2024-03-06	Modified the following content: <ul style="list-style-type: none"> Added Smart Connect APIs. Moved dumping APIs to Out-of-Date APIs.
2024-02-26	Modified the following content: <ul style="list-style-type: none"> Added the API for Producing Messages.
2023-11-10	Modified the following content: <ul style="list-style-type: none"> Added the Obtaining Instance Configurations and Modifying Instance Configurations APIs.
2023-07-26	Modified the following content: <ul style="list-style-type: none"> Added the Querying the Partition List of a Topic and Querying the Current Producer List of a Topic APIs.
2023-06-06	Modified the following content: <ul style="list-style-type: none"> Added the API for creating a consumer group.

Released On	Description
2023-04-23	<p>Modified the following content:</p> <ul style="list-style-type: none"> Added the API for deleting consumer groups in a Kafka instance in batches.
2023-02-03	<p>Modified the following content:</p> <ul style="list-style-type: none"> Updated the URIs in Creating an Instance, Modifying Instance Specifications, and Querying Product Information for Instance Specification Modification to support new instance specifications.
2022-06-30	<p>Modified the following content:</p> <ul style="list-style-type: none"> Added the description of new specifications to the specification parameter in Creating an instance. Added the broker_num parameter in Creating an instance.
2021-12-14	<p>Modified the following content:</p> <ul style="list-style-type: none"> Changed the APIs from V1 to V2 in Permissions and Supported Actions.
2021-11-16	<p>Modified the following content:</p> <ul style="list-style-type: none"> Added V2 APIs.
2020-10-13	<p>This issue is the first official release.</p>