

# Distributed Message Service for RabbitMQ

## API Reference

**Issue** 05  
**Date** 2023-05-08



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. 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

---

<b>1 Before You Start.....</b>	<b>1</b>
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Constraints.....	1
1.5 Concepts.....	2
<b>2 API Overview.....</b>	<b>4</b>
<b>3 Calling APIs.....</b>	<b>5</b>
3.1 Making an API Request.....	5
3.2 Authentication.....	9
3.3 Returned Values.....	11
<b>4 Getting Started.....</b>	<b>13</b>
<b>5 APIs V2 (Recommended).....</b>	<b>15</b>
5.1 Lifecycle Management.....	15
5.1.1 Creating an Instance.....	15
5.1.2 Listing All Instances.....	30
5.1.3 Querying an Instance.....	40
5.1.4 Deleting an Instance.....	48
5.1.5 Modifying Instance Information.....	51
5.1.6 Batch Deleting Instances.....	62
5.2 Instance Management.....	69
5.2.1 Resetting the Password.....	69
5.2.2 Listing Plug-ins.....	73
5.2.3 Enabling or Disabling a Plug-in.....	77
5.3 Specification Modification Management.....	81
5.3.1 Querying Product Information for Specification Modification of Instances with New Flavors.....	81
5.3.2 Modifying Specifications of Instances with New Flavors.....	88
5.4 Background Task Management.....	105
5.4.1 Listing Background Tasks.....	105
5.4.2 Querying a Background Task.....	110
5.4.3 Deleting a Background Task.....	114
5.5 Tag Management.....	117

5.5.1 Batch Adding or Deleting Tags.....	117
5.5.2 Listing Tags of an Instance.....	119
5.5.3 Listing Tags of a Project.....	123
5.6 Other APIs.....	127
5.6.1 Listing Maintenance Time Windows.....	127
5.6.2 Listing AZ Information.....	131
5.6.3 Querying Product Specifications.....	133
5.6.4 Querying Instance Monitoring Dimensions.....	139
<b>6 Permissions and Supported Actions.....</b>	<b>144</b>
<b>7 Out-of-Date APIs.....</b>	<b>148</b>
7.1 APIs V1.....	148
7.1.1 APIs for Managing Instances.....	148
7.1.1.1 Creating an Instance.....	148
7.1.1.2 Querying an Instance.....	153
7.1.1.3 Modifying an Instance.....	159
7.1.1.4 Deleting an Instance.....	162
7.1.1.5 Restarting or Deleting Instances in Batches.....	163
7.1.1.6 Querying All Instances.....	166
7.1.2 Other APIs.....	172
7.1.2.1 Querying AZ Information.....	172
7.1.2.2 Querying Product Specifications.....	174
7.1.2.3 Querying Maintenance Time Windows.....	185
<b>8 Appendix.....</b>	<b>188</b>
8.1 Status Code.....	188
8.2 Error Codes.....	191
8.3 Instance Status.....	214
8.4 Obtaining a Project ID.....	214
8.5 Obtaining the Account Name and Account ID.....	215
<b>A Change History.....</b>	<b>217</b>

# 1 Before You Start

---

## 1.1 Overview

Welcome to *Distributed Message Service (DMS) for RabbitMQ API Reference*. Distributed Message Service (DMS) for RabbitMQ is a message middleware service using the distributed, high-availability clustering technology. It provides reliable, scalable, and fully managed queues for sending, receiving, and storing messages.

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

---

### NOTICE

DMS for RabbitMQ is continuously upgraded with new functions, and the existing APIs are inevitably adjusted. For example, new response parameters are added.

To reduce the impact of API changes, DMS for RabbitMQ is backward compatible with existing APIs. When using DMS for RabbitMQ, you should accept and ignore unused parameters and parameter values in JSON responses.

---

## 1.2 API Calling

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

## 1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of CTS, see [Regions and Endpoints](#).

## 1.4 Constraints

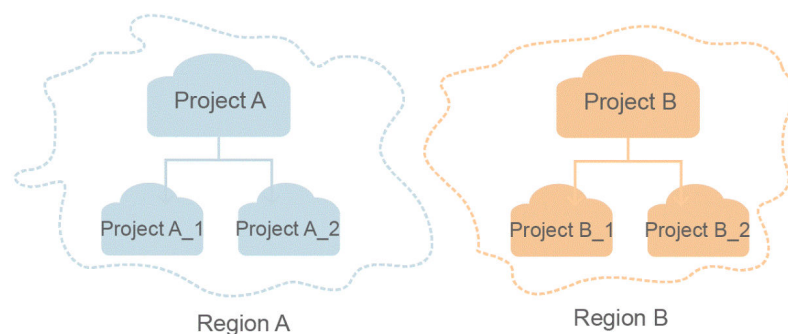
- The number of instances that you can create depends on your quota. To view or increase your quota, see [Quotas](#).

- For more constraints, see the API description.

## 1.5 Concepts

- Account  
An account is created upon successful registration and has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity and should not be used directly to perform routine management. To ensure security, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- IAM user  
An IAM user is created using an account to use cloud services. Each IAM user has its own identity credentials (password and access keys).  
The account name, username, and password will be required for API authentication.
- Region: Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.
- AZ: An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.
- Project  
A project corresponds to a region. Projects group and isolate resources (including compute, storage, and network resources) across physical regions. Users can be granted permissions in a default project to access all resources in the region associated with the 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 enterprise projects are logically isolated from each other. An enterprise project can contain resources in multiple regions, and resources can be directly transferred between 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** Instance management APIs

API	Description
<b>Lifecycle management APIs</b>	Create an instance, list all instances, query an instance, delete an instance, modify instance information, and restart or delete instances in batches.
<b>Instance management APIs</b>	Reset the password, list plug-ins, and enable or disable a plug-in.
<b>Specification modification management APIs</b>	Query product information for instance specification modification, and modify instance specifications.
<b>Background task management APIs</b>	List background tasks, query a background task, and delete a background task.
<b>Tag management APIs</b>	Add or delete tags in batches, list tags of an instance, and list tags of a project.
<b>Other APIs</b>	List maintenance time windows, AZ information, and product specifications.



# 3 Calling APIs

## 3.1 Making an API Request

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

### Request URI

A request URI is in the following format:

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

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

**Table 3-1** Parameters in a URI

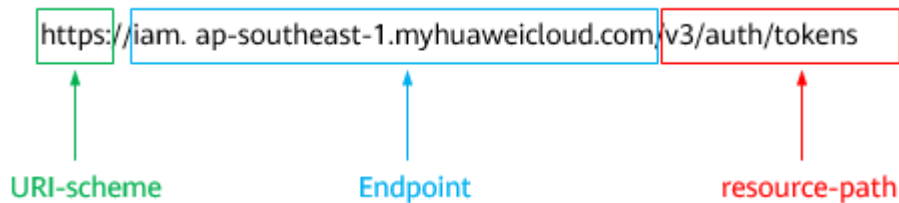
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from <b>Regions and Endpoints</b> . For example, the endpoint of IAM in the <b>CN-Hong Kong</b> region is <b>iam.ap-southeast-1.myhuaweicloud.com</b> .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to <b>obtain a user token</b> is <b>/v3/auth/tokens</b> .

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of <i>Parameter name=Parameter value</i> . For example, <b>?limit=10</b> indicates that a maximum of 10 data records will be displayed.

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

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

**Figure 3-1** Example URI



**NOTE**

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** 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.

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

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

## Request Header

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

**Table 3-2** 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 <b>application/json</b> 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 <a href="#">Obtaining a Project ID</a> .	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 to <b>obtain a user token</b>. This API is the only one that does not require authentication.</p> <p>After the request is processed, the value of <b>X-Subject-Token</b> in the response header is the token value.</p>	<p>No</p> <p>Mandatory for token-based authentication.</p>	<p>The following is part of an example token:</p> <p>MIIpAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ</p>

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

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

### 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.

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace **username**, **domainname**, **\*\*\*\*\*** (login password), and **xxxxxxxxxxxxxxxxxxxx** (project name) with the actual values (for example, ap-southeast-1). The project name can be obtained from [Regions and Endpoints](#).

 NOTE

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

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

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

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

## 3.2 Authentication

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

- Token-based authentication: Requests are authenticated using a token.
- AK/SK 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

 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 to [obtain a user token](#), you must set **auth.scope** in the request body to **project**.

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

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

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

## AK/SK-based Authentication

### NOTE

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

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

### 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 [obtaining a user token](#), in which **x-subject-token** is the desired user token. Then, you can use the token to authenticate the calling of other APIs.

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

```

connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

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

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → noopen

x-frame-options → SAMEORIGIN

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

x-subject-token
→ MIiYXQYJKoZIhvcNAQcCoIIYJCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjwMTktMDItMTNUMD.
fj3Kjs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JlGzrpd18LGXK5tdfq4lqHCYb8P4NaYONyejcAgzJVeFYtLWT1GSO0zxKZmLQHJQ82HBqHdglZO9fuEbl5dMhdavj+33wEI
xHRCe9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXlIjipPEGA270g1FruooL6jqglFKNPQuFSOU8+uSsttVwRtnfsC+qTp22Rkd5MCqFGQ8LcuUx3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbvpGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

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

#### (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.

You can obtain a token by **calling an API**. The following shows part of the response body for the API to obtain a user token.

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

```

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

## Creating a RabbitMQ Instance

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

```
{
  "name": "rabbitmq",
  "engine": "rabbitmq",
  "engine_version": "3.8.35",
  "storage_space": 100,
  "access_user": "test",
  "password": "ZxxxA",
  "vpc_id": "eadxxx72c",
  "security_group_id": "aa75axxc8c73220",
  "subnet_id": "3cb6axxx671d6a8",
  "available_zones": [
    "effdcxxb42f56533"
  ],
  "product_id": "c6.2u4g.single",
  "storage_spec_code": "dms.physical.storage.ultra.v2"
}
```

- **name**: name of the instance.
- **engine**: message engine. The value is **rabbitmq**.
- **engine\_version**: version of the message engine.
- **storage\_space**: message storage space in GB. For details about the value range, see [Creating an Instance](#).
- **access\_user**: user-defined username for logging in to RabbitMQ.
- **password**: user-defined password for logging in to RabbitMQ.

- **vpc\_id**: ID of the VPC where the RabbitMQ 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 Lifecycle Management

### 5.1.1 Creating an Instance

#### Function

This API is used to create an instance in the pay-per-use or yearly/monthly mode.

#### 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. Default: <b>rabbitmq</b>
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .

## 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. <b>NOTE</b> 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: <b>rabbitmq</b> .
engine_version	Yes	String	Version of the message engine. <ul style="list-style-type: none"> <li>The RabbitMQ version can be 3.8.35.</li> </ul>
storage_space	Yes	Integer	Message storage space in GB. Value range: <ul style="list-style-type: none"> <li>Single-node RabbitMQ instance: 100–90,000 GB</li> <li>Cluster RabbitMQ instance: 100 GB × Number of nodes to 90,000 GB, 200 GB × Number of nodes to 90,000 GB, and 300 GB × Number of nodes to 90,000 GB</li> </ul>
access_user	Yes	String	A username must start with a letter and only letters, digits, hyphens (-), and underscores (_) are allowed. It can contain 4 to 64 characters.

Parameter	Mandatory	Type	Description
password	Yes	String	Instance password. The password must meet the following complexity requirements: <ul style="list-style-type: none"> <li>• Contains 8 to 32 characters.</li> <li>• Contains at least two of the following character types: <ul style="list-style-type: none"> <li>- Lowercase letters</li> <li>- Uppercase letters</li> <li>- Digits</li> <li>- Special characters `~!@#\$%^&amp;*()-_+= []{}!'"&lt;.&gt;/?</li> </ul> </li> </ul>
vpc_id	Yes	String	VPC ID. To obtain it, log in to the VPC console and view the VPC ID on the VPC details page.
security_group_id	Yes	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.
subnet_id	Yes	String	Subnet ID. To obtain it, log in to VPC console and click the target subnet on the <b>Subnets</b> 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 <a href="#">Listing AZ Information</a> . This parameter cannot be empty or null.
product_id	Yes	String	Product ID. Obtain a product ID from <a href="#">Querying Product Specifications List</a> . If the product is of the cluster type (that is, <b>type</b> is set to <b>cluster</b> ), the <b>broker_num</b> field is mandatory.

Parameter	Mandatory	Type	Description
broker_num	No	Integer	<p>Number of brokers.</p> <p>If the product is of the single-node type, there can be only 1 broker. If the product is of the cluster type, there can be 3, 5, or 7 brokers.</p> <p>If the product type is <b>single</b>:</p> <ul style="list-style-type: none"> <li>• 1</li> </ul> <p>If the product type is <b>cluster</b>:</p> <ul style="list-style-type: none"> <li>• 3</li> <li>• 5</li> <li>• 7</li> </ul>
maintain_begin	No	String	<p>Indicates the time at which a maintenance time window starts. Format: HH:mm.</p> <ul style="list-style-type: none"> <li>• The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window.</li> <li>• The start time must be set to 22:00, 02:00, 06:00, 10:00, 14:00, or 18:00.</li> <li>• The start time and end time must be set in pairs. If the start time is left blank, the end time must also be left blank. In this case, the system automatically sets the start time to 02:00.</li> </ul>

Parameter	Mandatory	Type	Description
maintain_end	No	String	<p>Indicates the time at which a maintenance time window ends. Format: HH:mm.</p> <ul style="list-style-type: none"> <li>• The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window.</li> <li>• The end time is four hours later than the start time. For example, if the start time is 22:00, the end time is 02:00.</li> <li>• The start time and end time must be set in pairs. If the end time is left blank, the start time is also left blank. In this case, the system automatically sets the end time to 06:00.</li> </ul>
enable_publicip	No	Boolean	<p>Whether to enable public access for the RabbitMQ instance.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: enable</li> <li>• <b>false</b>: disable</li> </ul>
publicip_id	No	String	<p>ID of the EIP bound to the RabbitMQ instance. This parameter is mandatory if public access is enabled (that is, <b>enable_publicip</b> is set to <b>true</b>).</p>
ssl_enable	No	Boolean	<p>Whether to enable SSL encryption for access.</p> <ul style="list-style-type: none"> <li>• <b>true</b>: enable</li> <li>• <b>false</b>: disable</li> </ul>

Parameter	Mandatory	Type	Description
storage_spec_code	Yes	String	Storage I/O specification. Value range: <ul style="list-style-type: none"> <li>• <b>dms.physical.storage.high.v2</b></li> <li>• <b>dms.physical.storage.ultra.v2</b></li> <li>• <b>dms.physical.storage.high.dss.v2 (DeC)</b></li> <li>• <b>dms.physical.storage.ultra.dss.v2 (DeC)</b></li> </ul>
enterprise_project_id	No	String	Enterprise project ID. This parameter is mandatory for an enterprise project account.
tags	No	Array of <b>TagEntity</b> objects	Tag list.
bss_param	No	<b>BssParam</b> 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** TagEntity

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



Parameter	Mandatory	Type	Description
value	No	String	<p>Tag value.</p> <ul style="list-style-type: none"> <li>• Can contain 0 to 255 characters.</li> <li>• Can contain letters, digits, spaces, and special characters <code>._:=-@</code></li> <li>• Cannot start or end with a space.</li> </ul>

**Table 5-4** BssParam

Parameter	Mandatory	Type	Description
is_auto_renew	No	Boolean	<p>Whether auto renewal is enabled.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>true</b>: Auto renewal is enabled.</li> <li>• <b>false</b>: Auto renewal is not enabled.</li> </ul> <p>By default, auto renewal is disabled.</p>
charging_mode	No	String	<p>Billing mode.</p> <p>This parameter specifies a payment mode.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>prePaid</b>: yearly/monthly billing.</li> <li>• <b>postPaid</b>: pay-per-use billing.</li> </ul> <p>The default value is <b>postPaid</b>.</p>
is_auto_pay	No	Boolean	<p>Specifies whether the order is automatically or manually paid.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>true</b>: The order will be automatically paid.</li> <li>• <b>false</b>: The order must be manually paid.</li> </ul> <p>The default payment mode is manual.</p>

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

## Response Parameters

Status code: 200

Table 5-5 Response body parameters

Parameter	Type	Description
instance_id	String	Instance ID.

## Example Requests

- Creating a pay-per-use RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

POST [https://{endpoint}/v2/{engine}/{project\\_id}/instances](https://{endpoint}/v2/{engine}/{project_id}/instances)

```
{
  "name": "rabbitmq-demo",
  "description": "",
  "engine": "RabbitMQ",
  "engine_version": "3.8.35",
  "storage_space": 100,
  "access_user": "*****",
  "password": "*****",
  "vpc_id": "1e93f86e-13af-46c8-97d6-d40fa62b76c2",
  "security_group_id": "0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
  "subnet_id": "b5fa806c-35e7-4299-b659-b39398dd4718",
  "available_zones": [ "d573142f24894ef3bd3664de068b44b0" ],
```

```
"product_id" : "c6.2u4g.single",
"ssl_enable" : false,
"enable_publicip" : false,
"publicip_id" : "",
"storage_spec_code" : "dms.physical.storage.high.v2"
}
```

- Creating a yearly/monthly RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

POST https://{endpoint}/v2/{engine}/{project\_id}/instances

```
{
  "name" : "rabbitmq-demo",
  "description" : "",
  "engine" : "RabbitMQ",
  "engine_version" : "3.8.35",
  "storage_space" : 100,
  "access_user" : "*****",
  "password" : "*****",
  "vpc_id" : "1e93f86e-13af-46c8-97d6-d40fa62b76c2",
  "security_group_id" : "0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
  "subnet_id" : "b5fa806c-35e7-4299-b659-b39398dd4718",
  "available_zones" : [ "d573142f24894ef3bd3664de068b44b0" ],
  "product_id" : "c6.2u4g.single",
  "ssl_enable" : false,
  "enable_publicip" : false,
  "publicip_id" : "",
  "storage_spec_code" : "dms.physical.storage.high.v2",
  "bss_param" : {
    "charging_mode" : "prePaid",
    "period_type" : "month",
    "period_num" : 1,
    "is_auto_pay" : true
  }
}
```

## 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 RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;
```

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

public class CreatePostPaidInstanceByEngineSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        CreatePostPaidInstanceByEngineRequest request = new
        CreatePostPaidInstanceByEngineRequest();

        request.withEngine(CreatePostPaidInstanceByEngineRequest.EngineEnum.fromValue("{engine}"));
        CreateInstanceReq body = new CreateInstanceReq();
        List<String> listbodyAvailableZones = new ArrayList<>();
        listbodyAvailableZones.add("d573142f24894ef3bd3664de068b44b0");

        body.withStorageSpecCode(CreateInstanceReq.StorageSpecCodeEnum.fromValue("dms.physical.storag
        e.high.v2"));
        body.withSslEnable(false);
        body.withPublicIpId("");
        body.withEnablePublicIp(false);
        body.withProductId("c6.2u4g.single");
        body.withAvailableZones(listbodyAvailableZones);
        body.withSubnetId("b5fa806c-35e7-4299-b659-b39398dd4718");
        body.withSecurityGroupId("0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8");
        body.withVpId("1e93f86e-13af-46c8-97d6-d40fa62b76c2");
        body.withPassword("*****");
        body.withAccessUser("*****");
        body.withStorageSpace(100);
        body.withEngineVersion(CreateInstanceReq.EngineVersionEnum.fromValue("3.8.35"));
        body.withEngine(CreateInstanceReq.EngineEnum.fromValue("RabbitMQ"));
        body.withDescription("");
        body.withName("rabbitmq-demo");
        request.withBody(body);
        try {
            CreatePostPaidInstanceByEngineResponse response =
            client.createPostPaidInstanceByEngine(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 RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;

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

public class CreatePostPaidInstanceByEngineSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        CreatePostPaidInstanceByEngineRequest request = new
        CreatePostPaidInstanceByEngineRequest();

        request.withEngine(CreatePostPaidInstanceByEngineRequest.EngineEnum.fromValue("{engine}"));
        CreateInstanceReq body = new CreateInstanceReq();
        BssParam bssParambody = new BssParam();
        bssParambody.withChargingMode(BssParam.ChargingModeEnum.fromValue("prePaid"))
            .withIsAutoPay(true)
            .withPeriodType(BssParam.PeriodTypeEnum.fromValue("month"))
            .withPeriodNum(1);
        List<String> listbodyAvailableZones = new ArrayList<>();
        listbodyAvailableZones.add("d573142f24894ef3bd3664de068b44b0");
        body.withBssParam(bssParambody);

        body.withStorageSpecCode(CreateInstanceReq.StorageSpecCodeEnum.fromValue("dms.physical.storage
        e.high.v2"));
        body.withSslEnable(false);
        body.withPublicId("");
        body.withEnablePublicip(false);
        body.withProductId("c6.2u4g.single");
        body.withAvailableZones(listbodyAvailableZones);
        body.withSubnetId("b5fa806c-35e7-4299-b659-b39398dd4718");
        body.withSecurityGroupId("0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8");
        body.withVpcId("1e93f86e-13af-46c8-97d6-d40fa62b76c2");
        body.withPassword("*****");
        body.withAccessUser("*****");
        body.withStorageSpace(100);
        body.withEngineVersion(CreateInstanceReq.EngineVersionEnum.fromValue("3.8.35"));
        body.withEngine(CreateInstanceReq.EngineEnum.fromValue("RabbitMQ"));
        body.withDescription("");
        body.withName("rabbitmq-demo");
```

```
request.withBody(body);
try {
    CreatePostPaidInstanceByEngineResponse response =
client.createPostPaidInstanceByEngine(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 RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = CreatePostPaidInstanceByEngineRequest()
        request.engine = "{engine}"
        listAvailableZonesbody = [
            "d573142f24894ef3bd3664de068b44b0"
        ]
        request.body = CreateInstanceReq(
            storage_spec_code="dms.physical.storage.high.v2",
            ssl_enable=False,
            publicip_id="",
            enable_publicip=False,
            product_id="c6.2u4g.single",
            available_zones=listAvailableZonesbody,
            subnet_id="b5fa806c-35e7-4299-b659-b39398dd4718",
            security_group_id="0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
            vpc_id="1e93f86e-13af-46c8-97d6-d40fa62b76c2",
            password="*****",
            access_user="*****",
            storage_space=100,
            engine_version="3.8.35",
```

```
        engine="RabbitMQ",
        description="",
        name="rabbitmq-demo"
    )
    response = client.create_post_paid_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 RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

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

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = CreatePostPaidInstanceByEngineRequest()
        request.engine = "{engine}"
        bssParambody = BssParam(
            charging_mode="prePaid",
            is_auto_pay=True,
            period_type="month",
            period_num=1
        )
        listAvailableZonesbody = [
            "d573142f24894ef3bd3664de068b44b0"
        ]
        request.body = CreateInstanceReq(
            bss_param=bssParambody,
            storage_spec_code="dms.physical.storage.high.v2",
            ssl_enable=False,
            publicip_id="",
            enable_publicip=False,
            product_id="c6.2u4g.single",
            available_zones=listAvailableZonesbody,
            subnet_id="b5fa806c-35e7-4299-b659-b39398dd4718",
            security_group_id="0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
            vpc_id="1e93f86e-13af-46c8-97d6-d40fa62b76c2",
            password="*****",
            access_user="*****",
            storage_space=100,
            engine_version="3.8.35",
            engine="RabbitMQ",
            description="",
            name="rabbitmq-demo"
```

```
)
response = client.create_post_paid_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 RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePostPaidInstanceByEngineRequest{}
    request.Engine = model.GetCreatePostPaidInstanceByEngineRequestEngineEnum().ENGINE
    var listAvailableZonesbody = []string{
        "d573142f24894ef3bd3664de068b44b0",
    }
    sslEnableCreateInstanceReq:= false
    publicIpIdCreateInstanceReq:= ""
    enablePublicIpCreateInstanceReq:= false
    descriptionCreateInstanceReq:= ""
    request.Body = &model.CreateInstanceReq{
        StorageSpecCode:
model.GetCreateInstanceReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_HIGH,
        SslEnable: &sslEnableCreateInstanceReq,
        PublicIpId: &publicIpIdCreateInstanceReq,
        EnablePublicIp: &enablePublicIpCreateInstanceReq,
        ProductId: "c6.2u4g.single",
        AvailableZones: listAvailableZonesbody,
        SubnetId: "b5fa806c-35e7-4299-b659-b39398dd4718",
        SecurityGroupId: "0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
        VpId: "1e93f86e-13af-46c8-97d6-d40fa62b76c2",
        Password: "*****",
        AccessUser: "*****",
```



```

StorageSpace: int32(100),
EngineVersion: model.GetCreateInstanceReqEngineVersionEnum().E_3_8_35,
Engine: model.GetCreateInstanceReqEngineEnum().RABBIT_MQ,
Description: &descriptionCreateInstanceReq,
Name: "rabbitmq-demo",
}
response, err := client.CreatePostPaidInstanceByEngine(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

- Creating a yearly/monthly RabbitMQ instance whose version is 3.8.35, specifications are 2 vCPUs | 4 GB x 1, and storage space is 100 GB

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePostPaidInstanceByEngineRequest{
        request.Engine = model.GetCreatePostPaidInstanceByEngineRequestEngineEnum().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,
        }
        var listAvailableZonesbody = []string{
            "d573142f24894ef3bd3664de068b44b0",
        }
        sslEnableCreateInstanceReq:= false
        publicIpCreateInstanceReq:= ""
        enablePublicIpCreateInstanceReq:= false
        descriptionCreateInstanceReq:= ""
        request.Body = &model.CreateInstanceReq{

```

```

        BssParam: bssParambody,
        StorageSpecCode:
model.GetCreateInstanceReqStorageSpecCodeEnum().DMS_PHYSICAL_STORAGE_HIGH,
        SslEnable: &sslEnableCreateInstanceReq,
        PublicIpId: &publicIpIdCreateInstanceReq,
        EnablePublicIp: &enablePublicIpCreateInstanceReq,
        ProductId: "c6.2u4g.single",
        AvailableZones: listAvailableZonesbody,
        SubnetId: "b5fa806c-35e7-4299-b659-b39398dd4718",
        SecurityGroupId: "0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
        VpId: "1e93f86e-13af-46c8-97d6-d40fa62b76c2",
        Password: "*****",
        AccessUser: "*****",
        StorageSpace: int32(100),
        EngineVersion: model.GetCreateInstanceReqEngineVersionEnum().E_3_8_35,
        Engine: model.GetCreateInstanceReqEngineEnum().RABBIT_MQ,
        Description: &descriptionCreateInstanceReq,
        Name: "rabbitmq-demo",
    }
    response, err := client.CreatePostPaidInstanceByEngine(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 5-6** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .

**Table 5-7** Query Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine type. The value is <b>rabbitmq</b> . Default: <b>rabbitmq</b>
name	No	String	Instance name.
instance_id	No	String	Instance ID.
status	No	String	Instance status. For details, see <a href="#">Instance Status</a> .
include_failures	No	String	Whether to return the number of instances that fail to be created.  If the value is <b>true</b> , the number of instances that failed to be created is returned. If the value is not <b>true</b> , the number is not returned.
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 <b>true</b> , 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.

Parameter	Mandatory	Type	Description
limit	No	String	Maximum number of instances returned in the current query. The default value is <b>10</b> . The value ranges from 1 to 50.

## Request Parameters

None

## Response Parameters

Status code: 200

**Table 5-8** Response body parameters

Parameter	Type	Description
instances	Array of <a href="#">ShowInstanceResp</a> objects	Instance list.
instance_num	Integer	Number of instances.

**Table 5-9** ShowInstanceResp

Parameter	Type	Description
access_user	String	The username must be 4 to 64 characters long and can contain only letters, digits, and hyphens (-).
broker_num	Integer	Number of brokers.
name	String	Instance name.
engine	String	Message engine.
engine_version	String	Version of the message engine.
specification	String	Instance specifications. <ul style="list-style-type: none"> <li>For a single-node RabbitMQ instance, VM specifications are returned.</li> <li>For a cluster RabbitMQ instance, VM specifications and the number of nodes are returned.</li> </ul>
storage_space	Integer	Message storage space in GB.

Parameter	Type	Description
used_storage_space	Integer	Used message storage space in GB.
connect_address	String	IP address of an instance.
port	Integer	Port of an instance.
status	String	Instance status.
description	String	Instance description
instance_id	String	Instance ID.

Parameter	Type	Description
resource_spec_code	String	Resource specifications. <ul style="list-style-type: none"> <li>• <b>dms.instance.rabbitmq.single.c3.2u4g:</b> single-node RabbitMQ instance, 2 vCPUs 4 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.4u8g:</b> single-node RabbitMQ instance, 4 vCPUs 8 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.8u16g:</b> single-node RabbitMQ instance, 8 vCPUs 16 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.16u32g:</b> single-node RabbitMQ instance, 16 vCPUs 32 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.3:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.5:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.7:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.3:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.5:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.7:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.3</b> : cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.5</b> : cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.7</b> : cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 7 nodes</li> </ul>
charging_mode	Integer	Billing mode. The value <b>1</b> means pay-per-use, and <b>0</b> means yearly/monthly.
vpc_id	String	VPC ID.
vpc_name	String	VPC name.

Parameter	Type	Description
created_at	String	Time when an 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.
user_id	String	User ID.
user_name	String	Username.
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 to enable public access for the RabbitMQ instance. <ul style="list-style-type: none"> <li>● <b>true</b>: enable</li> <li>● <b>false</b>: disable</li> </ul>
publicip_address	String	EIP bound to the RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
publicip_id	String	ID of the EIP bound to the RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
management_connect_address	String	Management address of a RabbitMQ instance.
ssl_enable	Boolean	Whether security authentication is enabled. <ul style="list-style-type: none"> <li>● <b>true</b>: enabled</li> <li>● <b>false</b>: disabled</li> </ul>
enterprise_project_id	String	Enterprise project ID.
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"> <li>● <b>true</b>: New instance, which allows dynamic disk capacity expansion without restarting the instance.</li> <li>● <b>false</b>: old instance</li> </ul>

Parameter	Type	Description
extend_times	Integer	Number of disk expansion times. If the value exceeds 20, disk expansion is no longer allowed.
type	String	Instance type. The value can be <b>cluster</b> .
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 nodes belong. The AZ ID is returned.
total_storage_space	Integer	Message storage space in GB.
storage_resource_id	String	Storage resource ID.
storage_spec_code	String	I/O specifications.
ipv6_enable	Boolean	Whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	IPv6 connection address.
tags	Array of <b>TagEntity</b> objects	Tag list.

**Table 5-10** TagEntity

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



Parameter	Type	Description
value	String	Tag value. <ul style="list-style-type: none"> <li>• Can contain 0 to 255 characters.</li> <li>• Can contain letters, digits, spaces, and special characters <code>_.:=+@</code></li> <li>• Cannot start or end with a space.</li> </ul>

## Example Requests

Listing all instances

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

## Example Responses

**Status code: 200**

All instances are listed successfully.

```
{
  "instances": [ {
    "name": "api-explorer",
    "engine": "rabbitmq",
    "port": 5672,
    "status": "RUNNING",
    "type": "single",
    "specification": "2vCPUs 4GB",
    "engine_version": "3.8.35",
    "connect_address": "192.168.0.74",
    "instance_id": "de873040-d661-4770-aa96-9329c71d7c8a",
    "resource_spec_code": "dms.instance.rabbitmq.single.c3.2u4g",
    "charging_mode": 1,
    "vpc_id": "40a6501e-85ca-4449-a0db-b8bc7f0cec28",
    "vpc_name": "vpc-a400",
    "created_at": "1590047080687",
    "product_id": "00300-30109-0--0",
    "security_group_id": "bfd68e26-f8ef-4a91-a373-0a8f5c198601",
    "security_group_name": "Sys-default",
    "subnet_id": "a7f9a564-30dd-4059-8124-364ca6554578",
    "available_zones": [ "9f1c5806706d4c1fb0eb72f0a9b18c77" ],
    "user_id": "3df5acbc24a54fadb62a043c9000a307",
    "user_name": "*****",
    "maintain_begin": "22:00:00",
    "maintain_end": "02:00:00",
    "storage_space": 88,
    "total_storage_space": 100,
    "used_storage_space": 4,
    "enable_publicip": false,
    "ssl_enable": false,
    "management_connect_address": "http://192.168.0.74:15672",
    "storage_resource_id": "52be287d-1d6a-4d30-937e-185b3f176fc4",
    "storage_spec_code": "dms.physical.storage.normal",
    "enterprise_project_id": "0",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    }, {
      "key": "key2",
      "value": "value2"
    } ],
    "is_logical_volume": true,
  } ]
}
```

```
"extend_times" : 0,
"ipv6_enable" : false,
"ipv6_connect_addresses" : [ ]
}],
"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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;

public class ListInstancesDetailsSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ListInstancesDetailsRequest request = new ListInstancesDetailsRequest();
        try {
            ListInstancesDetailsResponse response = client.listInstancesDetails(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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ListInstancesDetailsRequest()
        response = client.list_instances_details(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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListInstancesDetailsRequest{}
    response, err := client.ListInstancesDetails(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 instances are listed successfully.

## 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 5-11** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

### Request Parameters

None

## Response Parameters

Status code: 200

**Table 5-12** Response body parameters

Parameter	Type	Description
access_user	String	The username must be 4 to 64 characters long and can contain only letters, digits, and hyphens (-).
broker_num	Integer	Number of brokers.
name	String	Instance name.
engine	String	Message engine.
engine_version	String	Version of the message engine.
specification	String	Instance specifications. <ul style="list-style-type: none"> <li>For a single-node RabbitMQ instance, VM specifications are returned.</li> <li>For a cluster RabbitMQ instance, VM specifications and the number of nodes are returned.</li> </ul>
storage_space	Integer	Message storage space in GB.
used_storage_space	Integer	Used message storage space in GB.
connect_address	String	IP address of an instance.
port	Integer	Port of an instance.
status	String	Instance status.
description	String	Instance description
instance_id	String	Instance ID.

Parameter	Type	Description
resource_spec_code	String	Resource specifications. <ul style="list-style-type: none"> <li>• <b>dms.instance.rabbitmq.single.c3.2u4g</b>: single-node RabbitMQ instance, 2 vCPUs 4 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.4u8g</b>: single-node RabbitMQ instance, 4 vCPUs 8 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.8u16g</b>: single-node RabbitMQ instance, 8 vCPUs 16 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.16u32g</b>: single-node RabbitMQ instance, 16 vCPUs 32 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.3</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.5</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.7</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.3</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.5</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.7</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.3</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.5</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.7</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 7 nodes</li> </ul>
charging_mode	Integer	Billing mode. The value <b>1</b> means pay-per-use, and <b>0</b> means yearly/monthly.
vpc_id	String	VPC ID.
vpc_name	String	VPC name.

Parameter	Type	Description
created_at	String	Time when an 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.
user_id	String	User ID.
user_name	String	Username.
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 to enable public access for the RabbitMQ instance. <ul style="list-style-type: none"> <li>● <b>true</b>: enable</li> <li>● <b>false</b>: disable</li> </ul>
publicip_address	String	EIP bound to the RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
publicip_id	String	ID of the EIP bound to the RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
management_connect_address	String	Management address of a RabbitMQ instance.
ssl_enable	Boolean	Whether security authentication is enabled. <ul style="list-style-type: none"> <li>● <b>true</b>: enabled</li> <li>● <b>false</b>: disabled</li> </ul>
enterprise_project_id	String	Enterprise project ID.
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"> <li>● <b>true</b>: New instance, which allows dynamic disk capacity expansion without restarting the instance.</li> <li>● <b>false</b>: old instance</li> </ul>

Parameter	Type	Description
extend_times	Integer	Number of disk expansion times. If the value exceeds 20, disk expansion is no longer allowed.
type	String	Instance type. The value can be <b>cluster</b> .
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 nodes belong. The AZ ID is returned.
total_storage_space	Integer	Message storage space in GB.
storage_resource_id	String	Storage resource ID.
storage_spec_code	String	I/O specifications.
ipv6_enable	Boolean	Whether IPv6 is enabled.
ipv6_connect_addresses	Array of strings	IPv6 connection address.
tags	Array of <b>TagEntity</b> objects	Tag list.

**Table 5-13** TagEntity

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



Parameter	Type	Description
value	String	Tag value. <ul style="list-style-type: none"> <li>• Can contain 0 to 255 characters.</li> <li>• Can contain letters, digits, spaces, and special characters <code>_.:=+-@</code></li> <li>• Cannot start or end with a space.</li> </ul>

## Example Requests

Querying details of a specified instance

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

## Example Responses

**Status code: 200**

The specified instance is queried successfully.

- Successful.

```
{
  "name": "api-explorer",
  "engine": "rabbitmq",
  "port": 5672,
  "status": "RUNNING",
  "type": "single",
  "specification": "2vCPUs 4GB",
  "engine_version": "3.8.35",
  "connect_address": "192.168.0.74",
  "instance_id": "de873040-d661-4770-aa96-9329c71d7c8a",
  "resource_spec_code": "dms.instance.rabbitmq.single.c3.2u4g",
  "charging_mode": 1,
  "vpc_id": "40a6501e-85ca-4449-a0db-b8bc7f0cec28",
  "vpc_name": "vpc-a400",
  "created_at": "1590047080687",
  "product_id": "00300-30109-0--0",
  "security_group_id": "bfd68e26-f8ef-4a91-a373-0a8f5c198601",
  "security_group_name": "Sys-default",
  "subnet_id": "a7f9a564-30dd-4059-8124-364ca6554578",
  "available_zones": [ "9f1c5806706d4c1fb0eb72f0a9b18c77" ],
  "user_id": "3df5acbc24a54fadb62a043c9000a307",
  "user_name": "paas_dms_01",
  "maintain_begin": "22:00:00",
  "maintain_end": "02:00:00",
  "storage_space": 88,
  "total_storage_space": 100,
  "used_storage_space": 4,
  "enable_publicip": false,
  "ssl_enable": false,
  "management_connect_address": "http://192.168.0.74:15672",
  "storage_resource_id": "52be287d-1d6a-4d30-937e-185b3f176fc4",
  "storage_spec_code": "dms.physical.storage.normal",
  "enterprise_project_id": "0",
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ],
}
```

```
"is_logical_volume" : true,  
"extend_times" : 0,  
"ipv6_enable" : false,  
"ipv6_connect_addresses" : [ ],  
"broker_num" : 1,  
"access_user" : "root_01"  
}
```

## 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.rabbitmq.v2.region.RabbitMQRegion;  
import com.huaweicloud.sdk.rabbitmq.v2.*;  
import com.huaweicloud.sdk.rabbitmq.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);  
  
        RabbitMQClient client = RabbitMQClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(RabbitMQRegion.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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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	The specified instance is queried successfully.

## 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 5-14** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.value_of("<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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
    rabbitmq.RabbitMQClientBuilder().
        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 instance is deleted successfully.

## Error Codes

See [Error Codes](#).

## 5.1.5 Modifying Instance Information

### Function

This API is used to modify the name and description of an instance.

### Calling Method

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

### URI

PUT /v2/{project\_id}/instances/{instance\_id}

**Table 5-15** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

## Request Parameters

**Table 5-16** Request body parameters

Parameter	Mandatory	Type	Description
name	No	String	Instance name. An instance name consists of 4 to 64 characters including letters, digits, and hyphens (-) and must start with a letter.
description	No	String	Description of an instance. The description supports up to 1024 characters. <b>NOTE</b> 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 \".



Parameter	Mandatory	Type	Description
maintain_begin	No	String	<p>Time at which the maintenance window starts. The format is HH:mm:ss.</p> <ul style="list-style-type: none"> <li>The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window.</li> <li>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.</li> <li>The start time and end time must be set in pairs. If the start time is left blank, the end time must also be left blank. In this case, the system automatically sets the start time to 02:00:00.</li> </ul>
maintain_end	No	String	<p>Time at which the maintenance window ends. The format is HH:mm:ss.</p> <ul style="list-style-type: none"> <li>The start time and end time of the maintenance time window must indicate the time segment of a supported maintenance time window.</li> <li>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.</li> <li>The start time and end time must be set in pairs. If the end time is left blank, the start time is also left blank. In this case, the system automatically sets the end time to 06:00:00.</li> </ul>
security_group_id	No	String	<p>Security group ID.</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
enable_publicip	No	Boolean	Whether to enable public access for the RabbitMQ instance. <ul style="list-style-type: none"> <li>• <b>true</b>: enable</li> <li>• <b>false</b>: disable</li> </ul>
publicip_id	No	String	ID of the EIP bound to a RabbitMQ instance. This parameter is mandatory if public access is enabled (that is, <b>enable_publicip</b> is set to <b>true</b> ). Method: Log in to the Network Console, go to the <b>EIPs</b> page, click the specified EIP, and find the ID in the <b>Basic Information</b> area.
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": "rabbitmq-01",
  "description": "instance description"
}
```

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

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

```
{
  "name": "rabbitmq-01",
  "description": "instance description",
  "maintain_begin": "02:00:00",
  "maintain_end": "06:00:00"
}
```

- Enabling public network access

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

```
{
  "enable_publicip": true,
  "publicip_id": "32685c2b-xxxx-xxxx-86c6-a1902359xxxx"
}
```

## 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withDescription("instance description");
        body.withName("rabbitmq-01");
        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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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("rabbitmq-01");
        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());
        }
    }
}
```

- Enabling public network 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateInstanceRequest request = new UpdateInstanceRequest();
        request.withInstanceId("{instance_id}");
        UpdateInstanceReq body = new UpdateInstanceReq();
        body.withPublicIpId("32685c2b-xxxx-xxxx-86c6-a1902359xxxx");
        body.withEnablePublicIp(true);
        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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

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

try:
    request = UpdateInstanceRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdateInstanceReq(
        description="instance description",
        name="rabbitmq-01"
    )
    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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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="rabbitmq-01"
        )
        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)
```

- Enabling public network access

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = UpdateInstanceRequest()
        request.instance_id = "{instance_id}"
        request.body = UpdateInstanceReq(
            publicip_id="32685c2b-xxxx-xxxx-86c6-a1902359xxxx",
            enable_publicip=True
        )
        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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
    rabbitmq.RabbitMQClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateInstanceRequest{
    request.InstanceId = "{instance_id}"
    descriptionUpdateInstanceReq:= "instance description"
    nameUpdateInstanceReq:= "rabbitmq-01"
    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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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:= "rabbitmq-01"
        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)
}
}
```

- Enabling public network access

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateInstanceRequest{
        request.InstanceId = "{instance_id}"
        publicIpUpdateInstanceReq := "32685c2b-xxxx-xxxx-86c6-a1902359xxxx"
        enablePublicIpUpdateInstanceReq := true
        request.Body = &model.UpdateInstanceReq{
            PublicIpId: &publicIpUpdateInstanceReq,
            EnablePublicIp: &enablePublicIpUpdateInstanceReq,
        }
    }
    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 is modified successfully.

## Error Codes

See [Error Codes](#).

## 5.1.6 Batch Deleting Instances

### Function

This API is used to delete instances in batches.

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 5-17** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .

### Request Parameters

**Table 5-18** 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. Value: <b>delete</b> .

Parameter	Mandatory	Type	Description
all_failure	No	String	Whether to delete instances that fail to be created. If this parameter is set to <b>rabbitmq</b> , all instances that fail to be created are deleted. In this case, the <b>instances</b> parameter in the request can be empty.

## Response Parameters

Status code: 200

Table 5-19 Response body parameters

Parameter	Type	Description
results	Array of <b>results</b> objects	Result of instance modification.

Table 5-20 results

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

## Example Requests

- 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": "rabbitmq"
}
```

## Example Responses

**Status code: 200**

Instances deleted.

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

## SDK Sample Code

The SDK sample code is as follows.

### Java

- 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchRestartOrDeleteInstancesRequest request = new BatchRestartOrDeleteInstancesRequest();
        BatchRestartOrDeleteInstanceReq body = new BatchRestartOrDeleteInstanceReq();
        body.withAllFailure(BatchRestartOrDeleteInstanceReq.AllFailureEnum.fromValue("rabbitmq"));
        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

- Deleting instances in batches

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkrabbitmq.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")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId) \  
  
    client = RabbitMQClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(RabbitMQRegion.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  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkrabbitmq.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")  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId) \  

```

```
client = RabbitMQClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(RabbitMQRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = BatchRestartOrDeleteInstancesRequest()
    request.body = BatchRestartOrDeleteInstanceReq(
        all_failure="rabbitmq",
        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

- Deleting instances in batches

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchRestartOrDeleteInstancesRequest{}
    allFailureBatchRestartOrDeleteInstanceReq:=
    model.GetBatchRestartOrDeleteInstanceReqAllFailureEnum().RABBITMQ
    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	Instances deleted.



Status Code	Description
204	All RabbitMQ instances that fail to be created are deleted successfully.

## Error Codes

See [Error Codes](#).

# 5.2 Instance Management

## 5.2.1 Resetting the Password

### Function

This API is used to reset the password.

### Calling Method

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

### URI

POST /v2/{project\_id}/instances/{instance\_id}/password

**Table 5-21** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

## Request Parameters

Table 5-22 Request body parameters

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

## 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

### Resetting a password

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResetPasswordRequest{}
    request.InstanceId = "{instance_id}"
    newPasswordResetPasswordReq := "*****"
    request.Body = &model.ResetPasswordReq{
        NewPassword: &newPasswordResetPasswordReq,
    }
    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 Listing Plug-ins

### Function

This API is used to list plug-ins.

### Calling Method

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

### URI

GET /v2/{project\_id}/instances/{instance\_id}/rabbitmq/plugins

Table 5-23 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

### Request Parameters

None

### Response Parameters

Status code: 200

**Table 5-24** Response body parameters

Parameter	Type	Description
plugins	Array of <b>PluginEntity</b> objects	Plug-in information list.

**Table 5-25** PluginEntity

Parameter	Type	Description
running	Boolean	Whether the plug-in is running.
enable	Boolean	Whether the plug-in is enabled.
name	String	Plug-in name.
version	String	Plug-in version.

## Example Requests

Querying the plug-in list

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

## Example Responses

**Status code: 200**

The plug-ins are listed successfully.

```
{
  "plugins" : [ {
    "running" : true,
    "enable" : true,
    "name" : "rabbitmq_shovel",
    "version" : "3.8.35"
  }, {
    "running" : true,
    "enable" : true,
    "name" : "rabbitmq_consistent_hash_exchange",
    "version" : "3.8.35"
  }, {
    "running" : false,
    "enable" : false,
    "name" : "rabbitmq_federation",
    "version" : "3.8.35"
  } ]
}
```

## 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;

public class ListPluginsSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ListPluginsRequest request = new ListPluginsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ListPluginsResponse response = client.listPlugins(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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId) \

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

try:
    request = ListPluginsRequest()
    request.instance_id = "{instance_id}"
    response = client.list_plugins(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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPluginsRequest{}
    request.InstanceId = "{instance_id}"
    response, err := client.ListPlugins(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 plug-ins are listed successfully.

## Error Codes

See [Error Codes](#).

## 5.2.3 Enabling or Disabling a Plug-in

### Function

This API is used to enable or disable a plug-in.

### Calling Method

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

### URI

PUT /v2/{project\_id}/instances/{instance\_id}/rabbitmq/plugins

**Table 5-26** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

### Request Parameters

**Table 5-27** Request body parameters

Parameter	Mandatory	Type	Description
enable	No	Boolean	Whether to enable the plug-in.

Parameter	Mandatory	Type	Description
plugins	No	String	Plug-ins. Separated multiple plug-ins with commas (,).

## Response Parameters

**Status code: 200**

**Table 5-28** Response body parameters

Parameter	Type	Description
job_id	String	Background task ID.

## Example Requests

Enabling the rabbitmq\_federation and rabbitmq\_shovel plug-ins

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

```
{
  "enable" : true,
  "plugins" : "rabbitmq_federation,rabbitmq_shovel"
}
```

## Example Responses

**Status code: 200**

The plug-in is enabled or disabled successfully.

```
{
  "job_id" : "8abfa7b27437db8f01744ea8ad4f245e"
}
```

## SDK Sample Code

The SDK sample code is as follows.

### Java

Enabling the rabbitmq\_federation and rabbitmq\_shovel plug-ins

```
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;
```

```
public class UpdatePluginsSolution {
    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);

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

        UpdatePluginsRequest request = new UpdatePluginsRequest();
        request.withInstanceId("{instance_id}");
        UpdatePluginsReq body = new UpdatePluginsReq();
        body.withPlugins("rabbitmq_federation,rabbitmq_shovel");
        body.withEnable(true);
        request.withBody(body);
        try {
            UpdatePluginsResponse response = client.updatePlugins(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

### Enabling the rabbitmq\_federation and rabbitmq\_shovel plug-ins

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(RabbitMQRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = UpdatePluginsRequest()
    request.instance_id = "{instance_id}"
    request.body = UpdatePluginsReq(
        plugins="rabbitmq_federation,rabbitmq_shovel",
        enable=True
    )
    response = client.update_plugins(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 the rabbitmq\_federation and rabbitmq\_shovel plug-ins

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePluginsRequest{}
    request.InstanceId = "{instance_id}"
    pluginsUpdatePluginsReq := "rabbitmq_federation,rabbitmq_shovel"
    enableUpdatePluginsReq := true
    request.Body = &model.UpdatePluginsReq{
        Plugins: &pluginsUpdatePluginsReq,
        Enable: &enableUpdatePluginsReq,
    }
    response, err := client.UpdatePlugins(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 plug-in is enabled or disabled successfully.

## Error Codes

See [Error Codes](#).

# 5.3 Specification Modification Management

## 5.3.1 Querying Product Information for Specification Modification of Instances with New Flavors

### Function

This API is used to query the product information for specification modification of instances with new flavors.

### Calling Method

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

### URI

GET /v2/{engine}/{project\_id}/instances/{instance\_id}/extend

**Table 5-29** Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine, which is <b>rabbitmq</b> . Default: <b>rabbitmq</b>
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

**Table 5-30** Query Parameters

Parameter	Mandatory	Type	Description
type	No	String	Product edition. <ul style="list-style-type: none"> <li>● <b>advanced</b>: the premium edition</li> <li>● <b>platinum</b>: the platinum edition</li> <li>● <b>dec</b>: the dedicated cloud edition</li> <li>● <b>exp</b>: the experience edition</li> </ul>

## Request Parameters

None

## Response Parameters

**Status code: 200**

**Table 5-31** Response body parameters

Parameter	Type	Description
engine	String	Message engine.
versions	Array of strings	Versions supported by the message engine.
products	Array of <a href="#">RabbitMQExtendProductInfoEntity</a> objects	Product information for specification modification.

**Table 5-32** RabbitMQExtendProductInfoEntity

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.

Parameter	Type	Description
charging_mode	Array of strings	Supported billing modes.
ios	Array of <a href="#">RabbitMQExtendProductIosEntity</a> objects	Disk I/O information.
properties	<a href="#">RabbitMQExtendProductPropertiesEntity</a> object	Key-value pair of a feature.
available_zones	Array of strings	AZs where there are available resources.
unavailable_zones	Array of strings	AZs where resources are sold out.
support_features	Array of <a href="#">RabbitMQProductSupportFeaturesEntity</a> objects	Supported features.

**Table 5-33** RabbitMQExtendProductIosEntity

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 5-34** RabbitMQExtendProductPropertiesEntity

Parameter	Type	Description
max_broker	String	Maximum number of brokers.
max_storage_per_node	String	Maximum storage space of each broker. Unit: GB.
min_broker	String	Minimum number of brokers.

Parameter	Type	Description
min_storage_per_node	String	Minimum storage space of each broker. Unit: GB.
max_connection_per_broker	String	Maximum number of connections.
step_length	String	Increment.
product_alias	String	Alias of <b>product_id</b> .
max_queue_per_broker	String	Maximum number of queues.

**Table 5-35** RabbitMQProductSupportFeaturesEntity

Parameter	Type	Description
name	String	Feature name.
properties	Map<String,String>	Key-value pair of a feature.

## Example Requests

Querying product information for instance specification modification

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

## Example Responses

**Status code: 200**

Successfully queried the product information for instance specification modification.

```
{
  "engine": "rabbitmq",
  "versions": [ "3.8.35" ],
  "products": [ {
    "type": "single",
    "product_id": "c6.2u4g.single",
    "ecs_flavor_id": "c6.large.2",
    "arch_types": [ "X86" ],
    "charging_mode": [ "monthly", "hourly" ],
    "ios": [ {
      "io_spec": "dms.physical.storage.ultra.v2",
      "available_zones": [ "xxx" ],
      "type": "evs",
      "unavailable_zones": [ "xxx" ]
    }, {
      "io_spec": "dms.physical.storage.high.v2",
      "available_zones": [ "xxx" ],
      "type": "evs",
      "unavailable_zones": [ "xxx" ]
    }
  ],
  } ],
}
```



```
"support_features" : [ ],
"properties" : {
  "max_connection_per_broker" : "2000",
  "max_broker" : "1",
  "max_queue_per_broker" : "100",
  "max_storage_per_node" : "30000",
  "min_broker" : "1",
  "product_alias" : "rabbitmq.2u4g.single",
  "step_length" : "0",
  "min_storage_per_node" : "100"
},
"available_zones" : [ "xxx" ],
"unavailable_zones" : [ ]
}, {
  "type" : "single",
  "product_id" : "c6.4u8g.single",
  "ecs_flavor_id" : "c6.xlarge.2",
  "arch_types" : [ "X86" ],
  "charging_mode" : [ "monthly", "hourly" ],
  "ios" : [ {
    "io_spec" : "dms.physical.storage.high.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ "xxx" ]
  }, {
    "io_spec" : "dms.physical.storage.ultra.v2",
    "available_zones" : [ "xxx" ],
    "type" : "evs",
    "unavailable_zones" : [ "xxx" ]
  } ],
  "support_features" : [ ],
  "properties" : {
    "max_connection_per_broker" : "3000",
    "max_broker" : "1",
    "max_queue_per_broker" : "200",
    "max_storage_per_node" : "30000",
    "min_broker" : "1",
    "product_alias" : "rabbitmq.4u8g.single",
    "step_length" : "0",
    "min_storage_per_node" : "100"
  },
  "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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

RabbitMQClient client = RabbitMQClient.newBuilder()
    .withCredential(auth)
    .withRegion(RabbitMQRegion.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.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 huaweicloudsdrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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](#).

## 5.3.2 Modifying Specifications of Instances with New Flavors

### Function

This API is used to modify instance specifications.

**Currently, this API can only be used to modify specifications of pay-per-use instances.**

### Calling Method

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

### URI

POST /v2/{engine}/{project\_id}/instances/{instance\_id}/extend

**Table 5-36** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.
engine	Yes	String	Message engine, which is <b>rabbitmq</b> . Default: <b>rabbitmq</b>

## Request Parameters

Table 5-37 Request body parameters

Parameter	Mandatory	Type	Description
oper_type	Yes	String	Change type. Value range: <b>storage</b> : Expand the storage without changing the broker quantity. <b>horizontal</b> : Add brokers without resizing the storage space of each broker. <b>vertical</b> : 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 <b>oper_type</b> is set to <b>storage</b> or <b>horizontal</b> . Instance storage space = Number of brokers x Storage space of each broker. If <b>oper_type</b> is set to <b>storage</b> , the number of brokers remains unchanged, and the storage space of each broker must be expanded by at least 100 GB. If <b>oper_type</b> is set to <b>horizontal</b> , the storage space of each broker remains unchanged.
new_product_id	No	String	Flavor, such as <b>c6.8u16g.cluster</b> . This parameter is valid and mandatory when <b>oper_type</b> is set to <b>vertical</b> .
new_broker_num	No	Integer	This parameter is valid only when <b>oper_type</b> is set to <b>horizontal</b> .

Parameter	Mandatory	Type	Description
new_spec_code	No	String	Old flavor, such as <b>dms.instance.rabbitmq.cluster.c3.8u16g</b> . When <b>oper_type</b> is set to <b>horizontal</b> , the value is <b>dms.instance.rabbitmq.cluster.c3.8u16g.5</b> (5 is the broker quantity.)

## Response Parameters

Status code: 200

Table 5-38 Response body parameters

Parameter	Type	Description
job_id	String	ID of the specification modification task.

## Example Requests

- Expanding the storage space (pay-per-use, old flavors). **new\_spec\_code** is the original specification.

```
POST https://{endpoint}/{engine}/v2/{project_id}/instances/{instance_id}/extend
```

```
{
  "new_storage_space" : 600,
  "oper_type" : "storage",
  "new_spec_code" : "dms.instance.rabbitmq.cluster.c3.2u4g.3"
}
```

- Adding brokers (pay-per-use, old flavors). **new\_storage\_space** is the space of the original specification. **new\_spec\_code** is **dms.instance.rabbitmq.cluster.c3.2u4g.5** (5 is the broker quantity.)

```
POST https://{endpoint}/{engine}/v2/{project_id}/instances/{instance_id}/extend
```

```
{
  "new_storage_space" : 600,
  "oper_type" : "horizontal",
  "new_spec_code" : "dms.instance.rabbitmq.cluster.c3.2u4g.5"
}
```

- Increasing broker flavors (pay-per-use, old flavors). For **new\_spec\_code**, replace the original specification with the new one. For example, replace 2u4g with 4u8g.

```
POST https://{endpoint}/{engine}/v2/{project_id}/instances/{instance_id}/extend
```

```
{
  "new_storage_space" : 600,
  "oper_type" : "vertical",
  "new_spec_code" : "dms.instance.rabbitmq.cluster.c3.2u4g.5"
}
```

- Expanding the storage space (pay-per-use)

```
POST https://{endpoint}/{engine}/v2/{project_id}/instances/{instance_id}/extend
{
  "new_storage_space" : 600,
  "oper_type" : "storage"
}
```

- Adding brokers (pay-per-use)

```
POST https://{endpoint}/{engine}/v2/{project_id}/instances/{instance_id}/extend
{
  "oper_type" : "horizontal",
  "new_storage_space" : 500,
  "new_broker_num" : 5
}
```

- 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 are modified successfully.

```
{
  "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, old flavors). **new\_spec\_code** is the original specification.

```
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

RabbitMQClient client = RabbitMQClient.newBuilder()
    .withCredential(auth)
    .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
    .build();
ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
request.withInstanceId("{instance_id}");
request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
body.withNewSpecCode("dms.instance.rabbitmq.cluster.c3.2u4g.3");
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, old flavors). **new\_storage\_space** is the space of the original specification. **new\_spec\_code** is **dms.instance.rabbitmq.cluster.c3.2u4g.5** (5 is the broker quantity.)

```
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);
```



```
RabbitMQClient client = RabbitMQClient.newBuilder()
    .withCredential(auth)
    .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
    .build();
ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
request.withInstanceId("{instance_id}");
request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
body.withNewSpecCode("dms.instance.rabbitmq.cluster.c3.2u4g.5");
body.withNewStorageSpace(600);
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 broker flavors (pay-per-use, old flavors). For **new\_spec\_code**, replace the original specification with the new one. For example, replace 2u4g with 4u8g.

```
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withInstanceId("{instance_id}");
```

```
request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
body.withNewSpecCode("dms.instance.rabbitmq.cluster.c3.2u4g.5");
body.withNewStorageSpace(600);
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());
}
}
```

- 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withInstanceId("{instance_id}");
        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
        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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withInstanceId("{instance_id}");
        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
        ResizeEngineInstanceReq body = new ResizeEngineInstanceReq();
        body.withNewBrokerNum(5);
        body.withNewStorageSpace(500);
        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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ResizeEngineInstanceRequest request = new ResizeEngineInstanceRequest();
        request.withInstanceId("{instance_id}");
        request.withEngine(ResizeEngineInstanceRequest.EngineEnum.fromValue("{engine}"));
        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, old flavors). **new\_spec\_code** is the original specification.

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

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ResizeEngineInstanceRequest()
        request.instance_id = "{instance_id}"
        request.engine = "{engine}"
        request.body = ResizeEngineInstanceReq(
            new_spec_code="dms.instance.rabbitmq.cluster.c3.2u4g.3",
            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, old flavors). **new\_storage\_space** is the space of the original specification. **new\_spec\_code** is **dms.instance.rabbitmq.cluster.c3.2u4g.5** (5 is the broker quantity.)

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
```

```
request = ResizeEngineInstanceRequest()
request.instance_id = "{instance_id}"
request.engine = "{engine}"
request.body = ResizeEngineInstanceReq(
    new_spec_code="dms.instance.rabbitmq.cluster.c3.2u4g.5",
    new_storage_space=600,
    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 broker flavors (pay-per-use, old flavors). For **new\_spec\_code**, replace the original specification with the new one. For example, replace 2u4g with 4u8g.

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

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"
```

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

```
    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
```

```
    request = ResizeEngineInstanceRequest()
    request.instance_id = "{instance_id}"
    request.engine = "{engine}"
    request.body = ResizeEngineInstanceReq(
        new_spec_code="dms.instance.rabbitmq.cluster.c3.2u4g.5",
        new_storage_space=600,
        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)
```

- Expanding the storage space (pay-per-use)

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

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ResizeEngineInstanceRequest()
        request.instance_id = "{instance_id}"
        request.engine = "{engine}"
        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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ResizeEngineInstanceRequest()
        request.instance_id = "{instance_id}"
        request.engine = "{engine}"
        request.body = ResizeEngineInstanceReq(
            new_broker_num=5,
            new_storage_space=500,
            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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ResizeEngineInstanceRequest()
        request.instance_id = "{instance_id}"
        request.engine = "{engine}"
        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, old flavors). **new\_spec\_code** is the original specification.

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
    rabbitmq.RabbitMQClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResizeEngineInstanceRequest{}
request.InstanceId = "{instance_id}"
request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
newSpecCodeResizeEngineInstanceReq:= "dms.instance.rabbitmq.cluster.c3.2u4g.3"
newStorageSpaceResizeEngineInstanceReq:= int32(600)
request.Body = &model.ResizeEngineInstanceReq{
    NewSpecCode: &newSpecCodeResizeEngineInstanceReq,
    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, old flavors). **new\_storage\_space** is the space of the original specification. **new\_spec\_code** is **dms.instance.rabbitmq.cluster.c3.2u4g.5** (5 is the broker quantity.)

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```

        WithCredential(auth).
        Build()

    request := &model.ResizeEngineInstanceRequest{}
    request.InstanceId = "{instance_id}"
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    newSpecCodeResizeEngineInstanceReq:= "dms.instance.rabbitmq.cluster.c3.2u4g.5"
    newStorageSpaceResizeEngineInstanceReq:= int32(600)
    request.Body = &model.ResizeEngineInstanceReq{
        NewSpecCode: &newSpecCodeResizeEngineInstanceReq,
        NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
        OperType: "horizontal",
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

- Increasing broker flavors (pay-per-use, old flavors). For **new\_spec\_code**, replace the original specification with the new one. For example, replace 2u4g with 4u8g.

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{}
    request.InstanceId = "{instance_id}"
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    newSpecCodeResizeEngineInstanceReq:= "dms.instance.rabbitmq.cluster.c3.2u4g.5"
    newStorageSpaceResizeEngineInstanceReq:= int32(600)
    request.Body = &model.ResizeEngineInstanceReq{
        NewSpecCode: &newSpecCodeResizeEngineInstanceReq,
        NewStorageSpace: &newStorageSpaceResizeEngineInstanceReq,
        OperType: "vertical",
    }
    response, err := client.ResizeEngineInstance(request)
    if err == nil {

```

```

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

```

- Expanding the storage space (pay-per-use)

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ResizeEngineInstanceRequest{}
    request.InstanceId = "{instance_id}"
    request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
    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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
    rabbitmq.RabbitMQClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ResizeEngineInstanceRequest{}
request.InstanceId = "{instance_id}"
request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
newBrokerNumResizeEngineInstanceReq:= int32(5)
newStorageSpaceResizeEngineInstanceReq:= int32(500)
request.Body = &model.ResizeEngineInstanceReq{
    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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```

Build()
request := &model.ResizeEngineInstanceRequest{}
request.InstanceId = "{instance_id}"
request.Engine = model.GetResizeEngineInstanceRequestEngineEnum().ENGINE
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 are modified successfully.

## Error Codes

See [Error Codes](#).

# 5.4 Background Task Management

## 5.4.1 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 5-39** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

**Table 5-40** 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 5-41** Response body parameters

Parameter	Type	Description
task_count	String	Number of tasks.
tasks	Array of <a href="#">tasks</a> objects	Task list.

**Table 5-42** tasks

Parameter	Type	Description
id	String	Task ID.

Parameter	Type	Description
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**

The tasks are listed successfully.

```
{
  "task_count" : "1",
  "tasks" : [ {
    "id" : "ff80808272dcc90f0172df1e490f41b0",
    "name" : "bindInstancePublicIp",
    "user_name" : "dms_test",
    "user_id" : "xxxxxxxx93ff484a828144c6xxxxxxxx",
    "params" : "{\"public_ip_id\":\"06a13350-4305-4338-9f0e-6b322bb1413d\", \"public_ip_address\": \"xx.xx.xx.xx\", \"enable_public_ip\": true}",
    "status" : "SUCCESS",
    "created_at" : "2020-06-23T03:00:03.471Z",
    "updated_at" : "2020-06-23T03:00:08.130Z"
  } ]
}
```

## 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

    RabbitMQClient client = RabbitMQClient.newBuilder()
        .withCredential(auth)
        .withRegion(RabbitMQRegion.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.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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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	The tasks are listed successfully.

## Error Codes

See [Error Codes](#).

## 5.4.2 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 5-43** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.
task_id	Yes	String	Task ID.

### Request Parameters

None

### Response Parameters

**Status code: 200**

**Table 5-44** Response body parameters

Parameter	Type	Description
task_count	String	Number of tasks.
tasks	Array of <a href="#">tasks</a> objects	Task list.

**Table 5-45** 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": "ff80808272dcc90f0172df1e490f41b0",
    "name": "bindInstancePublicIp",
    "user_name": "dms_test",
    "user_id": "xxxxxxxx93ff484a828144c6xxxxxxxx",
    "params": "{\"public_ip_id\":\"06a13350-4305-4338-9f0e-6b322bb1413d\", \"public_ip_address\": \"xx.xx.xx.xx\", \"enable_public_ip\": true}",
    "status": "SUCCESS",
    "created_at": "2020-06-23T03:00:03.471Z",
    "updated_at": "2020-06-23T03:00:08.130Z"
  } ]
}
```

## 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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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](#).

## 5.4.3 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 5-46** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.
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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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.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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

    client = RabbitMQClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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](#).

# 5.5 Tag Management

## 5.5.1 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}/rabbitmq/{instance\_id}/tags/action

**Table 5-47** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

## Request Parameters

**Table 5-48** Request body parameters

Parameter	Mandatory	Type	Description
action	No	String	Operation. Only lowercase letters are supported. <ul style="list-style-type: none"> <li>• <b>create</b>: Tags are created.</li> <li>• <b>delete</b>: Tags are deleted.</li> </ul>
tags	No	Array of <b>TagEntity</b> objects	Tag list.

**Table 5-49** TagEntity

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

## 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}/rabbitmq/{instance_id}/tags/action
{
  "action": "create",
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

## Example Responses

None

## Status Codes

Status Code	Description
204	Tags are successfully added or deleted.

## Error Codes

See [Error Codes](#).

## 5.5.2 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}/rabbitmq/{instance\_id}/tags

**Table 5-50** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see <a href="#">Obtaining a Project ID</a> .

Parameter	Mandatory	Type	Description
instance_id	Yes	String	Instance ID.

## Request Parameters

None

## Response Parameters

Status code: 200

Table 5-51 Response body parameters

Parameter	Type	Description
tags	Array of <b>TagEntity</b> objects	Tag list.

Table 5-52 TagEntity

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

## Example Requests

```
GET https://{endpoint}/v2/{project_id}/rabbitmq/{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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;

public class ShowRabbitMqTagsSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowRabbitMqTagsRequest request = new ShowRabbitMqTagsRequest();
        request.withInstanceId("{instance_id}");
        try {
            ShowRabbitMqTagsResponse response = client.showRabbitMqTags(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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId) \

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

    try:
        request = ShowRabbitMqTagsRequest()
        request.instance_id = "{instance_id}"
        response = client.show_rabbit_mq_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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```

Build()

request := &model.ShowRabbitMqTagsRequest{}
request.InstanceId = "{instance_id}"
response, err := client.ShowRabbitMqTags(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](#).

## 5.5.3 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}/rabbitmq/tags

**Table 5-53** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain it, see <a href="#">Obtaining a Project ID</a> .

## Request Parameters

None

## Response Parameters

Status code: 200

Table 5-54 Response body parameters

Parameter	Type	Description
tags	Array of <b>TagMultyValueEntity</b> objects	Tag list.

Table 5-55 TagMultyValueEntity

Parameter	Type	Description
key	String	Tag key.
values	Array of strings	Tag value

## Example Requests

```
GET https://{endpoint}/v2/{project_id}/rabbitmq/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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.v2.model.*;

public class ShowRabbitMqProjectTagsSolution {

    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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowRabbitMqProjectTagsRequest request = new ShowRabbitMqProjectTagsRequest();
        try {
            ShowRabbitMqProjectTagsResponse response = client.showRabbitMqProjectTags(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 huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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")
    projectId = "{project_id}"

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

```
client = RabbitMQClient.new_builder() \  
  .with_credentials(credentials) \  
  .with_region(RabbitMQRegion.value_of("<YOUR REGION>")) \  
  .build()  
  
try:  
  request = ShowRabbitMqProjectTagsRequest()  
  response = client.show_rabbit_mq_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"  
  rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"  
  "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"  
  region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(  
    rabbitmq.RabbitMQClientBuilder().  
      WithRegion(region.ValueOf("<YOUR REGION>")).  
      WithCredential(auth).  
      Build())  
  
  request := &model.ShowRabbitMqProjectTagsRequest{}  
  response, err := client.ShowRabbitMqProjectTags(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](#).

# 5.6 Other APIs

## 5.6.1 Listing Maintenance Time Windows

### Function

The 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 5-56 Response body parameters

Parameter	Type	Description
maintain_windows	Array of <a href="#">MaintainWindowsEntity</a> objects	List of supported maintenance time windows.

**Table 5-57** MaintainWindowsEntity

Parameter	Type	Description
default	Boolean	Whether a 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**

Maintenance time window queried successfully.

```
{
  "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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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 = RabbitMQClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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	Maintenance time window queried successfully.

## Error Codes

See [Error Codes](#).

## 5.6.2 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 5-58** Response body parameters

Parameter	Type	Description
region_id	String	Region ID.
available_zones	Array of <a href="#">available_zones</a> objects	Array of AZs.

**Table 5-59** available\_zones

Parameter	Type	Description
soldOut	Boolean	Whether resources are sold out.

Parameter	Type	Description
id	String	AZ ID.
code	String	AZ code.
name	String	AZ name.
port	String	AZ port.
resource_availability	String	Whether there are available resources in the AZ.
default_az	Boolean	Whether the AZ is the default AZ.
remain_time	Long	Remaining time.
ipv6_enable	Boolean	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
  }
  ]
}
```

## Status Codes

Status Code	Description
200	The AZ information is queried successfully.



## Error Codes

See [Error Codes](#).

## 5.6.3 Querying Product Specifications

### Function

This API is used to query the product specifications.

### Calling Method

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

### URI

GET /v2/{engine}/products

**Table 5-60** Path Parameters

Parameter	Mandatory	Type	Description
engine	Yes	String	Message engine. Default: <b>rabbitmq</b>

**Table 5-61** Query Parameters

Parameter	Mandatory	Type	Description
product_id	No	String	Product ID.

### Request Parameters

None

### Response Parameters

**Status code: 200**

**Table 5-62** Response body parameters

Parameter	Type	Description
engine	String	Message engine of DMS.

Parameter	Type	Description
versions	Array of strings	Supported versions.
products	Array of <a href="#">ListEngineProductsEntity</a> objects	Product specification details.

**Table 5-63** 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. <ul style="list-style-type: none"> <li>● <b>monthly</b>: yearly/monthly</li> <li>● <b>hourly</b>: pay-per-use</li> </ul>
ios	Array of <a href="#">ListEngineIosEntity</a> objects	List of supported disk I/O types.
support_features	Array of objects	List of features supported by instances of the current specifications.
properties	<a href="#">ListEnginePropertiesEntity</a> object	Attribute of instances of the current specifications.

**Table 5-64** ListEngineIosEntity

Parameter	Type	Description
io_spec	String	Disk I/O code.
type	String	Disk type.
available_zones	Array of strings	AZs.

Parameter	Type	Description
unavailable_zones	Array of strings	Unavailable AZs.

**Table 5-65** ListEnginePropertiesEntity

Parameter	Type	Description
step_length	String	Node quantity increase step.
max_queue_per_broker	String	Maximum number of queues on each broker.
max_connection_per_broker	String	Maximum number of connections on each broker.
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 <b>product_id</b> .

## Example Requests

```
GET https://{endpoint}/v2/rabbitmq/products
```

## Example Responses

**Status code: 200**

Product specifications are queried successfully.

- Successful.

```
{
  "engine": "rabbitmq",
  "versions": [ "3.8.35" ],
```

```
"products": [ {
  "type": "single",
  "product_id": "c6.2u4g.single",
  "ecs_flavor_id": "c6.large.2",
  "billing_code": "dms.platinum.c6",
  "arch_types": [ "X86" ],
  "charging_mode": [ "monthly", "hourly" ],
  "ios": [ {
    "io_spec": "dms.physical.storage.ultra.v2",
    "type": "evs",
    "available_zones": [ "xxx" ],
    "unavailable_zones": [ "xxx" ]
  }, {
    "io_spec": "dms.physical.storage.high.v2",
    "type": "evs",
    "available_zones": [ "xxx" ],
    "unavailable_zones": [ "xxx" ]
  } ],
  "support_features": [ ],
  "properties": {
    "max_connection_per_broker": "2000",
    "max_broker": "1",
    "max_queue_per_broker": "100",
    "max_storage_per_node": "30000",
    "min_broker": "1",
    "step_length": "0",
    "min_storage_per_node": "200",
    "product_alias": "rabbitmq.2u4g.single"
  }
}, {
  "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" ],
    "unavailable_zones": [ "xxx" ]
  }, {
    "io_spec": "dms.physical.storage.ultra.v2",
    "type": "evs",
    "available_zones": [ "xxx" ],
    "unavailable_zones": [ "xxx" ]
  } ],
  "support_features": [ ],
  "properties": {
    "max_connection_per_broker": "4500",
    "max_broker": "7",
    "max_queue_per_broker": "400",
    "max_storage_per_node": "30000",
    "min_broker": "3",
    "step_length": "2",
    "min_storage_per_node": "100",
    "product_alias": "rabbitmq.4u8g.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.rabbitmq.v2.region.RabbitMQRegion;
import com.huaweicloud.sdk.rabbitmq.v2.*;
import com.huaweicloud.sdk.rabbitmq.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);

        RabbitMQClient client = RabbitMQClient.newBuilder()
            .withCredential(auth)
            .withRegion(RabbitMQRegion.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

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkrabbitmq.v2.region.rabbitmq_region import RabbitMQRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkrabbitmq.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 = RabbitMQClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(RabbitMQRegion.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"
    rabbitmq "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/rabbitmq/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 := rabbitmq.NewRabbitMQClient(
        rabbitmq.RabbitMQClientBuilder().
            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	Product specifications are queried successfully.

## Error Codes

See [Error Codes](#).

## 5.6.4 Querying Instance Monitoring Dimensions

### Function

This API is used to query instance monitoring dimensions.

### Calling Method

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

### URI

GET /v2/{project\_id}/instances/{instance\_id}/ces-hierarchy

**Table 5-66** Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details, see <a href="#">Obtaining a Project ID</a> .
instance_id	Yes	String	Instance ID.

### Request Parameters

None

### Response Parameters

Status code: 200

**Table 5-67** Response body parameters

Parameter	Type	Description
dimensions	Array of <a href="#">dimensions</a> objects	Monitoring dimensions.

Parameter	Type	Description
instance_ids	Array of <b>instance_ids</b> objects	Instance information.
nodes	Array of <b>nodes</b> objects	Node information.
queues	Array of <b>queues</b> objects	Queue information.
vhosts	Array of <b>vhosts</b> objects	Virtual host information.
exchanges	Array of <b>exchanges</b> objects	Exchange information.
groups	Array of <b>groups</b> objects	Consumer group information.

**Table 5-68** 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 <b>children</b> objects	List of secondary dimensions.

**Table 5-69** children

Parameter	Type	Description
name	String	Secondary dimension name.
metrics	Array of strings	Metrics on the secondary dimension.



Parameter	Type	Description
key_name	Array of strings	Key used for monitoring query.
dim_router	Array of strings	Monitoring dimension route.

**Table 5-70** instance\_ids

Parameter	Type	Description
name	String	Instance ID.

**Table 5-71** nodes

Parameter	Type	Description
name	String	Node name.
available_zone	String	AZ.

**Table 5-72** queues

Parameter	Type	Description
name	String	Queue name.
vhost	String	Corresponding virtual host.

**Table 5-73** vhosts

Parameter	Type	Description
name	String	Virtual host name.

**Table 5-74** exchanges

Parameter	Type	Description
name	String	Exchange name.
vhost	String	Corresponding virtual host.

**Table 5-75** groups

Parameter	Type	Description
name	String	Consumer group name.

## Example Requests

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

## Example Responses

**Status code: 200**

The query is successful.

```
{
  "dimensions": [ {
    "name": "rabbitmq_instance_id",
    "metrics": [ "connections", "channels", "queues", "consumers", "messages_ready",
"messages_unacknowledged", "publish", "deliver", "deliver_no_ack", "deliver_get", "instance_bytes_in_rate",
"instance_bytes_out_rate", "instance_disk_usage" ],
    "key_name": [ "instance_ids" ],
    "dim_router": [ "rabbitmq_instance_id" ]
  }, {
    "name": "rabbitmq_node",
    "metrics": [ "fd_used", "socket_used", "proc_used", "mem_used", "disk_free", "rabbitmq_alive",
"rabbitmq_disk_usage", "rabbitmq_cpu_usage", "rabbitmq_cpu_core_load", "rabbitmq_memory_usage",
"rabbitmq_disk_read_await", "rabbitmq_disk_write_await", "rabbitmq_node_bytes_in_rate",
"rabbitmq_node_bytes_out_rate", "rabbitmq_node_queues", "rabbitmq_memory_high_watermark",
"rabbitmq_disk_insufficient" ],
    "key_name": [ "nodes" ],
    "dim_router": [ "rabbitmq_instance_id", "rabbitmq_node" ]
  }, {
    "name": "rabbitmq_queue",
    "metrics": [ "queue_messages_unacknowledged", "queue_messages_ready" ],
    "key_name": [ "queues" ],
    "dim_router": [ "rabbitmq_instance_id", "rabbitmq_queue" ]
  } ],
  "instance_ids": [ {
    "name": "0e16280d-7451-4f5b-80fa-f210372ce657"
  } ],
  "nodes": [ {
    "name": "dms-vm-0e16280d-rabbitmq-0",
    "available_zone": "cn-north-7c"
  }, {
    "name": "dms-vm-0e16280d-rabbitmq-1",
    "available_zone": "cn-north-7c"
  }, {
    "name": "dms-vm-0e16280d-rabbitmq-2",
    "available_zone": "cn-north-7c"
  } ],
  "queues": [ {
    "name": "Vhost-17130843_Queue-21084756",
    "vhost": "default"
  } ],
  "vhosts": [ {
    "name": "default"
  } ],
  "exchanges": [ {
    "name": "direct_exchange",
    "vhost": "default"
  } ],
  "groups": [ ]
}
```

## Status Codes

Status Code	Description
200	The query is successful.

## Error Codes

See [Error Codes](#).

# 6 Permissions and Supported Actions

---

This chapter describes fine-grained permissions management for your RabbitMQ 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 any permissions assigned. You need to add a user to one or more groups, and assign permissions policies to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on cloud services based on the permissions.

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

## NOTE

You can use policies to allow or deny access to specific APIs.

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The permissions required for calling an API are determined by the actions supported by the API. Only users that have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user creates a RabbitMQ instance using an API, the user must have been granted permissions that allow the **dms:instance:create** action.

## Supported Actions

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

- **Permission:** A statement in a policy that allows or denies certain operations.
- **APIs:** REST APIs that can be called in a custom policy.
- **Actions:** Added to a custom policy to control permissions for specific operations.

- IAM projects or enterprise projects: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management. For details about the differences between IAM and enterprise projects, see [What Are the Differences Between IAM and Enterprise Management?](#)

DMS for RabbitMQ supports the following actions that can be defined in custom policies. Permissions must be obtained before calling DMS APIs. For details on how to obtain permissions, visit the [Identity and Access Management help center](#).

**Table 6-1** DMS for RabbitMQ actions

Permissions	APIs	Actions	IAM Projects	Enterprise Projects
Creating a Pay-per-Use Instance	POST /v2/{project_id}/instances	dms:instance:create	√	√
Deleting an Instance	DELETE /v2/{project_id}/instances/{instance_id}	dms:instance:delete	√	√
Modifying an Instance	PUT /v2/{project_id}/instances/{instance_id}	dms:instance:modify	√	√
Querying an Instance	GET /v2/{project_id}/instances/{instance_id}	dms:instance:get	√	√
Restarting or Deleting Instances in Batches	POST /v2/{project_id}/instances/action	Restart: dms:instance:modifyStatus  Delete: dms:instance:delete	√	√
Listing All Instances	GET /v2/{project_id}/instances	dms:instance:list	√	√
Listing Plugins	GET /v2/{project_id}/instances/{instance_id}/rabbitmq/plugins	dms:instance:list	√	√

Permissions	APIs	Actions	IAM Projects	Enterprise Projects
Enabling or Disabling a Plug-in	PUT /v2/{project_id}/instances/{instance_id}/rabbitmq/plugins	dms:instance:modify	√	√
Resetting a Password	POST /v2/{project_id}/instances/{instance_id}/password	dms:instance:resetAuthInfo	√	√
Modifying Instance Specifications	POST /v2/{project_id}/instances/{instance_id}/extend	dms:instance:scale	√	√
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	GET /v2/{project_id}/instances/{instance_id}/tasks/{task_id}	dms:instance:deleteBackgroundTask	√	√
Adding or Deleting Instance Tags in Batches	POST /v2/{project_id}/rabbitmq/{instance_id}/tags/action	dms:instance:modify	√	√
Listing Tags of an Instance	GET /v2/{project_id}/rabbitmq/{instance_id}/tags	dms:instance:get	√	√
Querying Project Tags	GET /v2/{project_id}/rabbitmq/tags	dms:instance:get	√	√

Permissions	APIs	Actions	IAM Projects	Enterprise Projects
Enabling or Disabling Public Access	PUT /v2/{project_id}/instances/{instance_id}	dms:instance:modify	√	√

# 7 Out-of-Date APIs

---

## 7.1 APIs V1

### 7.1.1 APIs for Managing Instances

#### 7.1.1.1 Creating an Instance

 NOTE

This API is an out-of-date version 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** Parameter description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

#### Request

##### Request parameters

[Table 7-2](#) describes the request parameters.



**Table 7-2** Parameter description

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 supports only letters, digits, and hyphens (-).
description	String	No	Indicates the description of an instance. It is a character string containing not more than 1024 characters. <b>NOTE</b> 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. Value: <b>rabbitmq</b>
engine_version	String	No	Indicates the version of the message engine.
storage_space	Integer	Yes	Indicates the message storage space. Unit: GB <ul style="list-style-type: none"> <li>Single-node RabbitMQ instance: 100–90,000 GB</li> <li>Cluster RabbitMQ instance: 100 GB x Number of nodes to 90,000 GB, 200 GB x Number of nodes to 90,000 GB, and 300 GB x Number of nodes to 90,000 GB</li> </ul>
access_user	String	Yes	Indicates a username. A username consists of 4 to 64 characters and supports only letters, digits, and hyphens (-).
password	String	Yes	Indicates an instance password. An instance password must meet the following complexity requirements: <ul style="list-style-type: none"> <li>Must be a string consisting of 8 to 32 characters.</li> <li>Must contain at least two of the following character types: <ul style="list-style-type: none"> <li>Lowercase letters</li> <li>Uppercase letters</li> <li>Digits</li> <li>Special characters `~!@#%&amp;*( )- _ = + \   [ { } ; : ' , &lt; . &gt; / ?</li> </ul> </li> </ul>

Parameter	Type	Mandatory	Description
vpc_id	String	Yes	Indicates the ID of a VPC.
security_group_id	String	Yes	Indicates the ID of a security group.
subnet_id	String	Yes	Indicates the ID of a subnet.
available_zones	Array	Yes	Indicates the ID of an AZ. The parameter value cannot be an empty array or an empty array. For details, see <a href="#">Querying AZ Information</a> .
product_id	String	Yes	Indicates the product ID. For details, see <a href="#">Querying Product Specifications</a> .
maintain_begin	String	No	Indicates the time at which a maintenance time window starts. Format: HH:mm. <ul style="list-style-type: none"> <li>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 <a href="#">Querying Maintenance Time Windows</a>.</li> <li>The start time must be set to 22:00, 02:00, 06:00, 10:00, 14:00, or 18:00.</li> <li>Parameters <b>maintain_begin</b> and <b>maintain_end</b> must be set in pairs. If parameter <b>maintain_begin</b> is left blank, parameter <b>maintain_end</b> is also left blank. In this case, the system automatically set the start time to 02:00.</li> </ul>

Parameter	Type	Mandatory	Description
maintain_end	String	No	<p>Indicates the time at which a maintenance time window ends.</p> <p>Format: HH:mm.</p> <ul style="list-style-type: none"> <li>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 <a href="#">Querying Maintenance Time Windows</a>.</li> <li>The end time is four hours later than the start time. For example, if the start time is 22:00, the end time is 02:00.</li> <li>Parameters <b>maintain_begin</b> and <b>maintain_end</b> must be set in pairs. If parameter <b>maintain_end</b> is left blank, parameter <b>maintain_start</b> is also left blank. In this case, the system automatically set the end time to 06:00.</li> </ul>
enable_publicip	Boolean	No	<p>Indicates whether to enable public access for a RabbitMQ instance.</p> <ul style="list-style-type: none"> <li><b>true</b>: enable</li> <li><b>false</b>: disable</li> </ul>
publicip_id	String	No	<p>Indicates the ID of the elastic IP address (EIP) bound to a RabbitMQ instance.</p> <p>This parameter is mandatory if public access is enabled (that is, <b>enable_publicip</b> is set to <b>true</b>).</p>
ssl_enable	Boolean	No	<p>Indicates whether to enable SSL-encrypted access.</p> <ul style="list-style-type: none"> <li><b>true</b>: enable</li> <li><b>false</b>: disable</li> </ul>
storage_spec_code	String	Yes	<p>Indicates storage I/O specification. For details on how to select a disk type, see <a href="#">Disk Types and Performance</a>.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>dms.physical.storage.normal</li> <li>dms.physical.storage.high</li> <li>dms.physical.storage.ultra</li> </ul>
enterprise_project_id	String	No	<p>Indicates the enterprise project ID.</p>

### Example request for creating a RabbitMQ instance

```
{
  "name": "rabbitmq-demo",
  "description": "",
  "engine": "RabbitMQ",
  "engine_version": "3.x.x",
  "storage_space": 100,
  "access_user": "*****",
  "password": "*****",
  "vpc_id": "1e93f86e-13af-46c8-97d6-d40fa62b76c2",
  "security_group_id": "0aaa0033-bf7f-4c41-a6c2-18cd04cad2c8",
  "subnet_id": "b5fa806c-35e7-4299-b659-b39398dd4718",
  "available_zones": ["d573142f24894ef3bd3664de068b44b0"],
  "product_id": "00300-30109-0--0",
  "maintain_begin": "22:00",
  "maintain_end": "02:00",
  "ssl_enable": false,
  "enable_publicip": false,
  "publicip_id": "",
  "enterprise_project_id": "0",
  "storage_spec_code": "dms.physical.storage.ultra"
}
```

## Response

### Response parameters

[Table 7-3](#) describes the response parameter.

**Table 7-3** Parameter description

Parameter	Type	Description
instance_id	String	Indicates the instance ID.

### Example response

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

## Status Code

[Table 7-4](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-4** Status code

Status Code	Description
200	The instance is created successfully.

### 7.1.1.2 Querying an Instance

 NOTE

This API is an out-of-date version 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 7-5](#) describes the parameters.

**Table 7-5** Parameter description

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**

None.

#### Response

**Response parameters**

[Table 7-6](#) describes the response parameters.

**Table 7-6** Parameter description

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.

Parameter	Type	Description
specification	String	Indicates the instance specification. <ul style="list-style-type: none"><li>• For a single-node RabbitMQ instance, VM specifications are returned.</li><li>• For a cluster RabbitMQ instance, VM specifications and the number of nodes are returned.</li></ul>
storage_space	Integer	Indicates the message storage space. Unit: GB
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 <a href="#">Instance Status</a> .
description	String	Indicates the description of the instance.
instance_id	String	Indicates the instance ID.

Parameter	Type	Description
resource_spec_code	String	<p>Indicates the resource specifications identifier.</p> <ul style="list-style-type: none"> <li>• <b>dms.instance.rabbitmq.single.c3.2u4g:</b> single-node RabbitMQ instance, 2 vCPUs 4 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.4u8g:</b> single-node RabbitMQ instance, 4 vCPUs 8 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.8u16g:</b> single-node RabbitMQ instance, 8 vCPUs 16 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.16u32g:</b> single-node RabbitMQ instance, 16 vCPUs 32 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.3:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.5:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.7:</b> cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.3:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.5:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.7:</b> cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.3:</b> cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.5:</b> cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.7:</b> cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 7 nodes</li> </ul>
type	String	<p>Indicates the instance type.</p> <p>Options:</p> <ul style="list-style-type: none"> <li>• <b>single:</b> single-node instance</li> <li>• <b>cluster:</b> cluster instance</li> </ul>

Parameter	Type	Description
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.
error_code	String	Indicates an error code returned when an instance fails to be created or its status is abnormal. For details about error codes, see <a href="#">Table 7-7</a> .
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 segment.
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 an order ID. This parameter has a value only when the billing mode is yearly/monthly.
maintain_begin	String	Indicates the time at which a maintenance time window starts. Format: HH:mm
maintain_end	String	Indicates the time at which a maintenance time window ends. Format: HH:mm



Parameter	Type	Description
enable_publicip	Boolean	Indicates whether to enable public access for a RabbitMQ instance. <ul style="list-style-type: none"> <li>• <b>true</b>: enable</li> <li>• <b>false</b>: disable</li> </ul>
publicip_addresses	String	Indicates the EIP bound to a RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
publicip_id	String	Indicates the ID of the EIP bound to a RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
management_connect_address	String	Indicates the management address of a RabbitMQ instance.
ssl_enable	Boolean	Indicates whether to enable security authentication. <ul style="list-style-type: none"> <li>• <b>true</b>: enable</li> <li>• <b>false</b>: disable</li> </ul>
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"> <li>• <b>true</b>: New instance, which allows dynamic disk capacity expansion without restarting the instance.</li> <li>• <b>false</b>: Old instance.</li> </ul>
extend_times	String	Indicates the number of disk expansion times. If it exceeds 20, the disk cannot be expanded.

**Table 7-7** Error code description

Error Code	Description
public.00.0001	Internal service error.
public.00.0002	Internal service error.
public.00.0003	Internal service error.
public.00.0004	Failed to create the VPC.
public.00.0005	Failed to create the security group.

Error Code	Description
public.00.0006	Failed to create the subnet.
public.00.0007	The subnet status is abnormal.
public.00.0008	Failed to create the ECS.
public.00.0009	Failed to create the ECS.
public.00.0010	Failed to create the ECS.
public.00.0011	Failed to bind an NIC to the ECS.
public.00.0013	Failed to start the ECS.
public.00.0014	Failed to start the ECS.
public.00.0015	Failed to stop the ECS.
public.00.0018	Failed to create the ECS because the ECS resource quota is insufficient.
public.00.0024	Failed to deploy the instance.
public.00.0025	Some nodes of the instance are faulty.
public.00.0042	Failed to connect to the instance.

### Example response

```
{
  "name": "dms-a11e",
  "engine": "rabbitmq",
  "engine_version": "3.x.x",
  "specification": "2vCPUs 4GB",
  "storage_space": 100,
  "used_storage_space": 50,
  "connect_address": "192.168.3.100",
  "port": 5672,
  "status": "RUNNING",
  "description": "Create a instance",
  "instance_id": "68d5745e-6af2-40e4-945d-fe449be00148",
  "resource_spec_code": "dms.instance.rabbitmq.single.c3.2u4g",
  "type": "single",
  "charging_mode": 1,
  "vpc_id": "27d99e17-42f2-4751-818f-5c8c6c03ff15",
  "vpc_name": "vpc_4944a40e-ac57-4f08-9d38-9786e2759458_192",
  "created_at": "1526367063931",
  "error_code": null,
  "product_id": "00300-30109-0--0",
  "security_group_id": "60ea2db8-1a51-4ab6-9e11-65b418c24583",
  "security_group_name": "sg_6379_4944a40e-ac57-4f08-9d38-9786e2759458",
  "subnet_id": "ec2f34b9-20eb-4872-85bd-bea9fc943128",
  "subnet_name": "subnet_az_7f336767-10ec-48a5-9ae8-9cacde119318",
  "subnet_cidr": "192.168.0.0/24",
  "available_zones": ["1d7b939b382c4c3bb3481a8ca10da785"],
  "user_id": "6d0977e4c9b74ae7b5a083a8d0d8fafa",
  "user_name": "aabb02",
  "access_user": "user",
  "order_id": "XXXXXXXXXX",
  "maintain_begin": "22:00",
  "maintain_end": "02:00",
}
```

```
"enable_publicip" : "true",
"publicip_id": "b7940732-11ef-459b-acab-cab0d26c74a3",
"publicip_address": "192.168.10.5",
"ssl_enable": false,
"management_connect_address": "http://192.168.0.177:9999"
}
```

## Status Code

[Table 7-8](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-8** Status code

Status Code	Description
200	Specified instance queried successfully.

### 7.1.1.3 Modifying an Instance

#### NOTE

This API is an out-of-date version 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 name and description of an instance.

## URI

PUT /v1.0/{project\_id}/instances/{instance\_id}

**Table 7-9** Parameter description

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 7-10](#) describes the request parameters.

**Table 7-10** Parameter description

Parameter	Type	Mandatory	Description
name	String	No	Indicates the instance name. An instance name is a string of 4 to 64 characters that contain letters, digits, and hyphens (-). An instance name must start with a letter.
description	String	No	Indicates the description of an instance. It is a character string containing not more than 1024 characters. <b>NOTE</b> 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 \".
maintain_begin	String	No	Indicates the time at which a maintenance time window starts. Format: HH:mm:ss. <ul style="list-style-type: none"> <li>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 <a href="#">Querying Maintenance Time Windows</a>.</li> <li>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.</li> <li>Parameters <b>maintain_begin</b> and <b>maintain_end</b> must be set in pairs. If parameter <b>maintain_begin</b> is left blank, parameter <b>maintain_end</b> is also left blank. In this case, the system automatically set the start time to 02:00:00.</li> </ul>

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"> <li>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 <a href="#">Querying Maintenance Time Windows</a>.</li> <li>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.</li> <li>Parameters <b>maintain_begin</b> and <b>maintain_end</b> must be set in pairs. If parameter <b>maintain_end</b> is left blank, parameter <b>maintain_start</b> is also left blank. In this case, the system automatically allocates the default end time 06:00:00.</li> </ul>
security_group_id	String	No	Indicates the security group ID.
enable_publicip	Boolean	No	<p>Indicates whether to enable public access for a RabbitMQ instance.</p> <ul style="list-style-type: none"> <li><b>true</b>: enable</li> <li><b>false</b>: disable</li> </ul>
publicip_id	String	No	<p>Indicates the ID of the EIP bound to a RabbitMQ instance.</p> <p>This parameter is mandatory if public access is enabled (that is, <b>enable_publicip</b> is set to <b>true</b>).</p>
enterprise_project_id	String	No	Indicates the enterprise project ID.

### Example request

Example 1:

```
{
  "name": "dms002",
```

```
"description": "instance description"
}
```

Example 2:

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

## Response

### Response parameters

None.

### Example response

None.

## Status Code

[Table 7-11](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-11** Status code

Status Code	Description
204	The instance is modified successfully.

### 7.1.1.4 Deleting an Instance

#### NOTE

This API is an out-of-date version 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 7-12](#) describes the parameters.

**Table 7-12** Parameter description

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

None.

## Response

### Response parameters

None.

### Example response

None.

## Status Code

**Table 7-13** describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-13** 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 an out-of-date version 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 7-14** describes the parameter.

**Table 7-14** Parameter description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.

## Request

Request

**Table 7-15** describes the request parameters.

**Table 7-15** Parameter description

Parameter	Type	Mandatory	Description
action	String	Yes	Indicates the operation to be performed on instances. The value of this parameter can be <b>restart</b> or <b>delete</b> .
instances	Array	Yes	Indicates the list of instance IDs.
allFailure	String	No	Indicates whether to delete instances that fail to be created in batches. If this parameter is set to <b>true</b> , all instances that fail to be created are deleted. In this case, the <b>instances</b> parameter in the request can be empty.

### Example request

Restarting instances in batches:

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

Deleting instances in batches:



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

Deleting all instances that fail to be created:

```
{
  "action": "delete",
  "allFailure": "true"
}
```

## Response

### Response parameters

When **action** is set to **delete**, **allFailure** is set to **true**, and an empty response is returned, the instances are deleted successfully. [Table 7-16](#) describes the parameters.

**Table 7-16** Parameter description

Parameter	Type	Description
results	Array	Indicates the result of instance modification.

**Table 7-17** results parameter description

Parameter	Type	Description
instance	String	Indicates the instance ID.
result	String	Indicates an operation result, which can be <b>success</b> or <b>failed</b>

### Example response

```
{
  "results": [
    {
      "result": "success",
      "instance": "afc90a2a-a02c-4cba-94d5-58dfa9ad1e0d"
    },
    {
      "result": "success",
      "instance": "67fc5f8d-3986-4f02-bb75-4075a23112de"
    }
  ]
}
```

## Status Code

[Table 7-18](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-18** Status code

Status Code	Description
200	The instances are restarted or deleted successfully.

### 7.1.1.6 Querying All Instances

 NOTE

This API is an out-of-date version 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 7-19](#) describes the parameters.

**Table 7-19** Parameter description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Indicates the ID of a project.
engine	String	No	Indicates a message engine type The value is <b>rabbitmq</b> . If this parameter is not specified, all instances will be queried.
name	String	No	Indicates the instance name.
id	String	No	Indicates the instance ID.
status	String	No	Indicates the instance status. For details, see <a href="#">Instance Status</a> .
includeFailure	String	No	Indicates whether to return the number of instances that fail to be created. If the value is <b>true</b> , the number of instances that failed to be created is returned. If the value is not <b>true</b> , 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 <b>false</b> , indicating that a fuzzy search is performed based on a specified instance name. If the value is <b>true</b> , the instance that precisely matches a specified instance name is queried.
enterprise_project_id	String	No	Indicates the enterprise project ID.

### Example

```
GET /v1.0/bd6b78e2ff9e4e47bc260803ddcc7a21/instances?
start=1&limit=10&name=&status=&id=&includeFailure=true&exactMatchName=false
```

## Request

### Request parameters

None.

### Example request

None.

## Response

### Response parameters

[Table 7-20](#) describes the response parameters.

**Table 7-20** Parameter description

Parameter	Type	Description
instances	Array	Indicates instance details.
instance_num	Integer	Indicates the number of instances.

**Table 7-21** 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.
specification	String	Indicates the specifications of an instance. <ul style="list-style-type: none"> <li>For a single-node RabbitMQ instance, VM specifications are returned.</li> <li>For a cluster RabbitMQ instance, VM specifications and the number of nodes are returned.</li> </ul>
storage_space	Integer	Indicates the message storage space. Unit: GB
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 <a href="#">Instance Status</a> .
description	String	Indicates the description of the instance.
instance_id	String	Indicates the instance ID.

Parameter	Type	Description
resource_spec_code	String	<p>Indicates the resource specifications identifier.</p> <ul style="list-style-type: none"> <li>• <b>dms.instance.rabbitmq.single.c3.2u4g</b>: single-node RabbitMQ instance, 2 vCPUs 4 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.4u8g</b>: single-node RabbitMQ instance, 4 vCPUs 8 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.8u16g</b>: single-node RabbitMQ instance, 8 vCPUs 16 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.single.c3.16u32g</b>: single-node RabbitMQ instance, 16 vCPUs 32 GB (VM specifications)</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.3</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.5</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.4u8g.7</b>: cluster RabbitMQ instance, 4 vCPUs 8 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.3</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.5</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.8u16g.7</b>: cluster RabbitMQ instance, 8 vCPUs 16 GB (VM specifications), 7 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.3</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 3 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.5</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 5 nodes</li> <li>• <b>dms.instance.rabbitmq.cluster.c3.16u32g.7</b>: cluster RabbitMQ instance, 16 vCPUs 32 GB (VM specifications), 7 nodes</li> </ul>
charging_mode	Integer	Indicates a billing mode. <b>1</b> : pay-per-use mode; <b>0</b> : yearly/monthly billing.
vpc_id	String	Indicates the ID of a VPC.
vpc_name	String	Indicates the name of a VPC.

Parameter	Type	Description
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.
error_code	String	Indicates an error code returned when an instance fails to be created or its status is abnormal. For details about error codes, see <a href="#">Table 7-7</a> .
user_id	String	Indicates the user ID.
user_name	String	Indicates the username.
order_id	String	Indicates an order ID. This parameter has a value only when the billing mode is yearly/monthly mode.
maintain_begin	String	Indicates the time at which a maintenance time window starts. Format: HH:mm
maintain_end	String	Time at which the maintenance time window ends. Format: HH:mm
enable_publicip	Boolean	Indicates whether to enable public access for a RabbitMQ instance. <ul style="list-style-type: none"> <li>● <b>true</b>: enable</li> <li>● <b>false</b>: disable</li> </ul>
publicip_addresses	String	Indicates the EIP bound to a RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
publicip_id	String	Indicates the ID of the EIP bound to a RabbitMQ instance. The value of this parameter is <b>null</b> if public access is disabled.
management_connect_address	String	Indicates the management address of a RabbitMQ instance.
ssl_enable	Boolean	Indicates whether to enable security authentication. <ul style="list-style-type: none"> <li>● <b>true</b>: enable</li> <li>● <b>false</b>: disable</li> </ul>
enterprise_project_id	String	Indicates the enterprise project ID.

Parameter	Type	Description
is_logical_volume	Boolean	Distinguishes old instances from new instances during instance capacity expansion. <ul style="list-style-type: none"> <li>● <b>true</b>: New instance, which allows dynamic disk capacity expansion without restarting the instance.</li> <li>● <b>false</b>: Old instance.</li> </ul>
extend_times	String	Indicates the number of disk expansion times. If it exceeds 20, the disk cannot be expanded.

### Example response

```
{
  "instances": [
    {
      "name": "rabbitmq-lxy001",
      "engine": "rabbitmq",
      "port": 5672,
      "status": "RUNNING",
      "type": "single",
      "specification": "2vCPUs 4GB",
      "engine_version": "3.x.x",
      "connect_address": "192.168.255.237",
      "instance_id": "595926bf-a648-47d8-91bc-461956794c2b",
      "resource_spec_code": "dms.instance.rabbitmq.single.c3.2u4g",
      "charging_mode": 1,
      "vpc_id": "1a28dcc5-c90d-421c-82bb-783f30f5b40a",
      "vpc_name": "vpc-y00292973",
      "created_at": "1562583302800",
      "product_id": "00300-30109-0--0",
      "security_group_id": "0cc8fdb7-872a-49da-a062-88ccc39463b5",
      "security_group_name": "sg-65eb-nw-test",
      "subnet_id": "ebba7994-260d-42ab-bce1-39a08b365dc8",
      "available_zones": [
        "d573142f24894ef3bd3664de068b44b0"
      ],
      "user_id": "50a4156d334a4a82b8745dc730dc1e00",
      "user_name": "test",
      "access_user": "test-Info",
      "maintain_begin": "02:00:00",
      "maintain_end": "06:00:00",
      "storage_space": 88,
      "total_storage_space": 100,
      "used_storage_space": 4,
      "enable_publicip": false,
      "ssl_enable": false,
      "management_connect_address": "http://192.168.255.237:15672",
      "storage_resource_id": "34825335-61cb-4ee0-949e-24b08170edb2",
      "storage_spec_code": "dms.physical.storage.ultra",
      "service_type": "advanced",
      "storage_type": "hec",
      "enterprise_project_id": "0",
      "is_logical_volume": false,
      "extend_times": 0,
      "ipv6_enable": false,
      "ipv6_connect_addresses": [],
      "connector_enable": false,
      "connector_id": "",
      "rest_enable": false,
      "rest_connect_address": "",
      "public_boundwidth": 0,
    }
  ]
}
```

```
    "message_query_inst_enable": true,  
    "vpc_client_plain": false,  
    "support_features":  
"feature.physerver.kafka.topic.accesspolicy,message_trace_enable,feature.physerver.kafka.pulbic.dynamic,feat  
ure.physerver.kafka.user.manager",  
    "trace_enable": false  
  }  
],  
  "instance_num": 1  
}
```

## Status Code

**Table 7-22** describes the status code of successful operations. For details about other status codes, see **Status Code**.

**Table 7-22** Status code

Status Code	Description
200	All instances are queried successfully.

## 7.1.2 Other APIs

### 7.1.2.1 Querying AZ Information

 **NOTE**

This API is an out-of-date version 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

None.

#### Response

##### Response parameters



[Table 7-23](#) and [Table 7-24](#) describe the parameters.

**Table 7-23** Response parameters

Parameter	Type	Description
regionId	String	Indicates the region ID.
available_zones	Array	Indicates details of AZs. For details, see <a href="#">Table 7-24</a> .

**Table 7-24** available\_zones parameter description

Parameter	Type	Description
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"> <li>• <b>true</b>: The AZ has available resources.</li> <li>• <b>false</b>: Resources of the AZ have been sold out.</li> </ul>

### Example response

```
{
  regionId: "XXXXXX",
  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 7-25](#) describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-25** Status code

Status Code	Description
200	The AZ information is successfully queried.

## 7.1.2.2 Querying Product Specifications

### NOTE

This API is an out-of-date version 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 ID (parameter **product\_id**) which indicates the specifications of the service you purchased.

## URI

GET /v1.0/products?engine={engine}

[Table 7-26](#) describes the parameter.

**Table 7-26** Parameter description

Parameter	Type	Mandatory	Description
engine	String	No	Indicates the message engine.

## Request

### Request parameters

None.

### Example Request

None.

## Response

### Response parameters

[Table 7-27](#) describes the response parameters.

**Table 7-27** Parameter description

Parameter	Type	Description
name	String	Indicates the name of a message engine.
version	String	Indicates the version of the message engine.
values	Array	Indicates product specifications.

**Table 7-28** values parameter description

Parameter	Type	Description
detail	Array	Indicates the details of specifications.
name	String	Indicates an instance type, which can be single-node or cluster.

**Table 7-29** detail parameter description of single-node RabbitMQ instances

Parameter	Type	Description
storage	String	Indicates the message storage space.
io	Array	Indicates the I/O information.
vm_specification	String	Indicates VM specifications.
product_id	String	Indicates the product ID.
spec_code	String	Indicates the specification ID.

**Table 7-30** detail parameter description of cluster RabbitMQ instances

Parameter	Type	Description
vm_specification	String	Indicates VM specifications.
product_info	Array	Indicates the product information.

**Table 7-31** product\_info parameter description

Parameter	Type	Description
storage	String	Indicates the message storage space.
io	Array	Indicates the I/O information.
node_num	Integer	Indicates the number of nodes in a cluster.

Parameter	Type	Description
product_id	String	Indicates the product ID.
spec_code	String	Indicates the specification ID.

**Table 7-32** io parameter description

Parameter	Type	Description
io_type	String	Indicates the I/O type.
storage_spec_code	String	Indicates the I/O specification.

### Example response

List of RabbitMQ instance specifications:

```
{
  "Hourly": [
    {
      "name": "RabbitMQ",
      "version": "3.x.x",
      "values": [
        {
          "detail": [
            {
              "storage": "100",
              "io": [
                {
                  "io_type": "normal",
                  "storage_spec_code": "dms.physical.storage.normal"
                },
                {
                  "io_type": "high",
                  "storage_spec_code": "dms.physical.storage.high"
                },
                {
                  "io_type": "ultra",
                  "storage_spec_code": "dms.physical.storage.ultra"
                }
              ],
              "vm_specification": "2vCPUs 4GB",
              "product_id": "00300-30109-0--0",
              "spec_code": "dms.instance.rabbitmq.single.c3.2u4g"
            },
            {
              "storage": "100",
              "io": [
                {
                  "io_type": "normal",
                  "storage_spec_code": "dms.physical.storage.normal"
                },
                {
                  "io_type": "high",
                  "storage_spec_code": "dms.physical.storage.high"
                },
                {
                  "io_type": "ultra",
                  "storage_spec_code": "dms.physical.storage.ultra"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "vm_specification": "4vCPUs 8GB",
  "product_id": "00300-30111-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.4u8g"
},
{
  "storage": "100",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "vm_specification": "8vCPUs 16GB",
  "product_id": "00300-30113-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.8u16g"
},
{
  "storage": "100",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "vm_specification": "16vCPUs 32GB",
  "product_id": "00300-30115-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.16u32g"
}
],
"name": "single"
},
{
  "detail": [
    {
      "vm_specification": "4vCPUs 8GB",
      "product_info": [
        {
          "storage": "300",
          "io": [
            {
              "io_type": "normal",
              "storage_spec_code": "dms.physical.storage.normal"
            },
            {
              "io_type": "high",
              "storage_spec_code": "dms.physical.storage.high"
            },
            {
              "io_type": "ultra",
              "storage_spec_code": "dms.physical.storage.ultra"
            }
          ]
        }
      ]
    }
  ]
}

```

```

    ],
    "node_num": "3",
    "product_id": "00300-30209-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.3"
  },
  {
    "storage": "500",
    "io": [
      {
        "io_type": "normal",
        "storage_spec_code": "dms.physical.storage.normal"
      },
      {
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra"
      }
    ],
    "node_num": "5",
    "product_id": "00300-30211-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.5"
  },
  {
    "storage": "700",
    "io": [
      {
        "io_type": "normal",
        "storage_spec_code": "dms.physical.storage.normal"
      },
      {
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra"
      }
    ],
    "node_num": "7",
    "product_id": "00300-30213-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.7"
  }
]
},
{
  "vm_specification": "8vCPUs 16GB",
  "product_info": [
    {
      "storage": "300",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "3",
      "product_id": "00300-30215-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.3"
    }
  ]
}

```

```

    },
    {
      "storage": "500",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "5",
      "product_id": "00300-30217-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.5"
    },
    {
      "storage": "700",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "7",
      "product_id": "00300-30219-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.7"
    }
  ]
},
{
  "vm_specification": "16vCPUs 32GB",
  "product_info": [
    {
      "storage": "300",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "3",
      "product_id": "00300-30221-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.3"
    },
    {
      "storage": "500",
      "io": [

```

```

        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "5",
      "product_id": "00300-30223-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.5"
    },
    {
      "storage": "700",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        },
        {
          "io_type": "high",
          "storage_spec_code": "dms.physical.storage.high"
        },
        {
          "io_type": "ultra",
          "storage_spec_code": "dms.physical.storage.ultra"
        }
      ],
      "node_num": "7",
      "product_id": "00300-30225-0--0",
      "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.7"
    }
  ]
},
"name": "cluster"
}
]
},
"Monthly": [
  {
    "name": "RabbitMQ",
    "version": "3.x.x",
    "values": [
      {
        "detail": [
          {
            "storage": "100",
            "io": [
              {
                "io_type": "normal",
                "storage_spec_code": "dms.physical.storage.normal"
              },
              {
                "io_type": "high",
                "storage_spec_code": "dms.physical.storage.high"
              },
              {
                "io_type": "ultra",
                "storage_spec_code": "dms.physical.storage.ultra"
              }
            ],
            "vm_specification": "2vCPUs 4GB",
          }
        ]
      }
    ]
  }
]

```



```

"product_id": "00300-30110-0--0",
"spec_code": "dms.instance.rabbitmq.single.c3.2u4g"
},
{
  "storage": "100",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "vm_specification": "4vCPUs 8GB",
  "product_id": "00300-30112-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.4u8g"
},
{
  "storage": "100",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "vm_specification": "8vCPUs 16GB",
  "product_id": "00300-30114-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.8u16g"
},
{
  "storage": "100",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "vm_specification": "16vCPUs 32GB",
  "product_id": "00300-30116-0--0",
  "spec_code": "dms.instance.rabbitmq.single.c3.16u32g"
}
],
"name": "single"
},
{
  "detail": [
    {

```

```

"vm_specification": "4vCPUs 8GB",
"product_info": [
  {
    "storage": "300",
    "io": [
      {
        "io_type": "normal",
        "storage_spec_code": "dms.physical.storage.normal"
      },
      {
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra"
      }
    ],
    "node_num": "3",
    "product_id": "00300-30210-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.3"
  },
  {
    "storage": "500",
    "io": [
      {
        "io_type": "normal",
        "storage_spec_code": "dms.physical.storage.normal"
      },
      {
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra"
      }
    ],
    "node_num": "5",
    "product_id": "00300-30212-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.5"
  },
  {
    "storage": "700",
    "io": [
      {
        "io_type": "normal",
        "storage_spec_code": "dms.physical.storage.normal"
      },
      {
        "io_type": "high",
        "storage_spec_code": "dms.physical.storage.high"
      },
      {
        "io_type": "ultra",
        "storage_spec_code": "dms.physical.storage.ultra"
      }
    ],
    "node_num": "7",
    "product_id": "00300-30214-0--0",
    "spec_code": "dms.instance.rabbitmq.cluster.c3.4u8g.7"
  }
],
},
{
  "vm_specification": "8vCPUs 16GB",
  "product_info": [
    {
      "storage": "300",

```

```

"io": [
  {
    "io_type": "normal",
    "storage_spec_code": "dms.physical.storage.normal"
  },
  {
    "io_type": "high",
    "storage_spec_code": "dms.physical.storage.high"
  },
  {
    "io_type": "ultra",
    "storage_spec_code": "dms.physical.storage.ultra"
  }
],
"node_num": "3",
"product_id": "00300-30216-0--0",
"spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.3"
},
{
  "storage": "500",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "node_num": "5",
  "product_id": "00300-30218-0--0",
  "spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.5"
},
{
  "storage": "700",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "node_num": "7",
  "product_id": "00300-30220-0--0",
  "spec_code": "dms.instance.rabbitmq.cluster.c3.8u16g.7"
}
]
},
{
  "vm_specification": "16vCPUs 32GB",
  "product_info": [
    {
      "storage": "300",
      "io": [
        {
          "io_type": "normal",
          "storage_spec_code": "dms.physical.storage.normal"
        }
      ]
    }
  ]
}

```

```

    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "node_num": "3",
  "product_id": "00300-30222-0--0",
  "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.3"
},
{
  "storage": "500",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "node_num": "5",
  "product_id": "00300-30224-0--0",
  "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.5"
},
{
  "storage": "700",
  "io": [
    {
      "io_type": "normal",
      "storage_spec_code": "dms.physical.storage.normal"
    },
    {
      "io_type": "high",
      "storage_spec_code": "dms.physical.storage.high"
    },
    {
      "io_type": "ultra",
      "storage_spec_code": "dms.physical.storage.ultra"
    }
  ],
  "node_num": "7",
  "product_id": "00300-30226-0--0",
  "spec_code": "dms.instance.rabbitmq.cluster.c3.16u32g.7"
}
]
}
],
"name": "cluster"
}
]
}
}

```

## Status Code

**Table 7-33** describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-33** Status code

Status Code	Description
200	Product specifications queried successfully.

### 7.1.2.3 Querying Maintenance Time Windows

 NOTE

This API is an out-of-date version 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

None.

#### Response

##### Response parameters

[Table 7-34](#) and [Table 7-35](#) describe the response parameters.

**Table 7-34** Parameter description

Parameter	Type	Description
maintain_windows	Array	Indicates a list of supported maintenance time windows.

**Table 7-35** maintain\_windows parameter description

Parameter	Type	Description
seq	Integer	Indicates the sequential number of a maintenance time window.

Parameter	Type	Description
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": [
    {
      "seq": 1,
      "begin": "22",
      "end": "02",
      "default": false
    },
    {
      "seq": 2,
      "begin": "02",
      "end": "06",
      "default": true
    },
    {
      "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",
      "default": false
    }
  ]
}
```

### Status Code

**Table 7-36** describes the status code of successful operations. For details about other status codes, see [Status Code](#).

**Table 7-36** Status code

<b>Status Code</b>	<b>Description</b>
200	The maintenance time windows are queried successfully.

# 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 <b>timeout</b> 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.
400	DMS.00400081	Duplicate instance name.	The instance name already exists.	Check the instance name.

Status Code	Error Codes	Error Message	Description	Solution
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.
400	DMS.00400101	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.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.0040010 2	The number of partitions created for a Kafka topic must be within the range of 1-20.	The number of partitions created for a topic in a Kafka instance must be within the range of 1-20.	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.
400	DMS.0040010 7	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.0040010 8	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.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.0040012 4	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.0040012 5	Invalid SPEC_CODE.	Invalid SPEC_CODE.	Check SPEC_CODE.
400	DMS.0040012 6	Invalid period time.	Invalid time period for yearly/ monthly billing.	Check the time period for yearly/ monthly billing.
400	DMS.0040012 7	Instance not support to change retention_policy.	The instance does not support retention policy changes.	Contact technical support.
400	DMS.0040012 8	Invalid public access parameters.	Invalid public access parameters.	Check the public access parameters.
400	DMS.0040012 9	Current instance version is less than required.	The instance version does not support this operation.	Contact technical support.
400	DMS.0040013 3	Sink task quota for connector invalid.	Invalid connector task quota.	Contact technical support.
400	DMS.0040013 4	There is another order need to pay first.	An unpaid order exists.	Pay for the order first.
400	DMS.0040013 5	Not support disk encrypted.	Disk encryption is not supported.	Do not enable disk encryption.
400	DMS.0040013 6	Disk encrypted key is null.	The disk encryption key is empty.	Check the disk encryption key.

Status Code	Error Codes	Error Message	Description	Solution
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.
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.

Status Code	Error Codes	Error Message	Description	Solution
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.
400	DMS.00404033	Does not support extend rabbitMQ disk space.	The RabbitMQ disk space cannot be expanded.	Scale out the RabbitMQ cluster.
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.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.1024000 2	The number of queried queues exceeds the upper limit.	The maximum number of queried queues has been reached.	Check the queue quantity.
400	DMS.1024000 4	The tag name is invalid.	Invalid tag name.	Check the tag name.
400	DMS.1024000 5	The project ID format is invalid.	Invalid project ID.	Check the project ID format.
400	DMS.1024000 7	The name contains invalid characters.	The name contains invalid characters.	Check the name.
400	DMS.1024000 9	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.
400	DMS.1024001 0	The description contains invalid characters.	The description contains invalid characters.	Check the description.
400	DMS.1024001 1	The name length must be 1 to 64 characters.	The name can contain 1 to 64 characters.	Check the name length.
400	DMS.1024001 2	The name length must be 1 to 32 characters.	The name can contain 1 to 32 characters.	Check the name length.
400	DMS.1024001 3	The description length must not exceed 160 characters.	The description can contain a maximum of 160 characters.	Check the description length.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.1024001 4	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.1024001 5	The queue ID format is invalid.	Invalid queue ID.	Check the queue ID.
400	DMS.1024001 6	The group ID format is invalid.	Invalid group ID.	Check the group ID.
400	DMS.1024001 7	The queue already exists.	The queue already exists.	Check whether the queue exists.
400	DMS.1024001 8	The consumer group already exists.	The consumer group already exists.	Check whether the consumer group exists.
400	DMS.1024001 9	The number of consumer groups exceeds the upper limit.	The number of consumer groups exceeds the upper limit.	Check the number of consumer groups.
400	DMS.1024002 0	The quota is insufficient.	Insufficient quota.	Check the quota.
400	DMS.1024002 1	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.1024002 2	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.1024002 7	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.



Status Code	Error Codes	Error Message	Description	Solution
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.
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.

Status Code	Error Codes	Error Message	Description	Solution
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.
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.

Status Code	Error Codes	Error Message	Description	Solution
400	DMS.1054001 2	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.1054001 4	The URL contains invalid parameters.	The URL contains invalid parameters.	Check the URL parameters.
400	DMS.1054020 2	The request format is incorrect	The request format is incorrect. {Error description}	Check the request format.
400	DMS.1054220 4	Failed to consume messages due to {desc}.	Failed to consume messages. {Error description}	Check the error information and rectify the fault accordingly.
400	DMS.1054220 5	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.

Status Code	Error Codes	Error Message	Description	Solution
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.
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.

Status Code	Error Codes	Error Message	Description	Solution
403	DMS.0040300 2	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.0040300 3	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.1024030 4	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.1024030 6	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.1024030 7	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.1024030 8	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.1024030 9	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.

Status Code	Error Codes	Error Message	Description	Solution
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.
403	DMS.10240312	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.00404001	The requested URL does not exist.	The requested URL does not exist.	Check the URL.
404	DMS.00404022	This instance does not exist.	The instance does not exist.	Check whether the instance exists.
404	DMS.00404024	Connector does not exist.	Connector does not exist.	Check the connector.
404	DMS.00404026	The dumping task does not exist.	The dumping task does not exist.	Check the dumping task.
404	DMS.00404027	Connector already exists.	Connector already exists.	Check the connector.
404	DMS.00404029	The dumping task quota has been reached.	The dumping task quota has been reached.	Check the dumping task quota.
404	DMS.10240401	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.

Status Code	Error Codes	Error Message	Description	Solution
404	DMS.10240405	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.10240406	The URL or endpoint does not exist.	The URL or endpoint does not exist.	Check whether the URL or endpoint exists and is correct.
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.

Status Code	Error Codes	Error Message	Description	Solution
500	DMS.0050005 2	Internal service error.	Failed to submit the instance upgrade job.	Contact technical support.
500	DMS.0050005 3	Internal service error.	The specified instance node is not found.	Contact technical support.
500	DMS.0050005 4	Internal service error.	Failed to generate the password.	Contact technical support.
500	DMS.0050007 0	Internal service error.	Failed to configure the instance.	Contact technical support.
500	DMS.0050007 1	Internal service error.	Failed to create the instance backup policy.	Contact technical support.
500	DMS.0050009 4	Internal service error.	Internal service error.	Contact technical support.
500	DMS.0050010 6	Internal service error.	Internal service error.	Contact technical support.
500	DMS.0050099 0	Failed to update topics.	Failed to update topics.	Contact technical support.
500	DMS.0050100 0	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.0050100 1	Failed to get agency roleId.	Failed to get agency roleId.	retry the request later
500	DMS.0050100 2	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.0050100 3	Failed to grant role to agency.	Failed to grant role to agency.	Try again later, or contact technical support



Status Code	Error Codes	Error Message	Description	Solution
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.
500	DMS.00501012	Smart Connect tasks exist.	Smart Connect tasks exist.	Delete all Smart Connect tasks.
500	DMS.10250002	Internal service error.	Internal service error.	Contact technical support.
500	DMS.10250003	Internal service error.	Internal service error.	Contact technical support.
500	DMS.10250004	Internal service error.	Internal service error.	Contact technical support.
500	DMS.10250005	Internal communication error.	Internal communication error.	Contact technical support.
500	DMS.10250006	Internal service error.	Internal service error.	Contact technical support.
500	DMS.10550035	tag_type must be either or or and.	tag_type must be either 'or' or 'and'.	Check tag_type.
501	DMS.111501026	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.111501025	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.
Faulty	The instance is not running properly.
Restarting	The instance is being restarted.
Starting	The status between <b>Frozen</b> and <b>Running</b> .
Changing	The instance specifications are being changed.
Change failed	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 <b>My Order</b> .
Freezing	The status between <b>Running</b> and <b>Frozen</b> .
Upgrading	The instance is being upgraded.
Rolling back	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

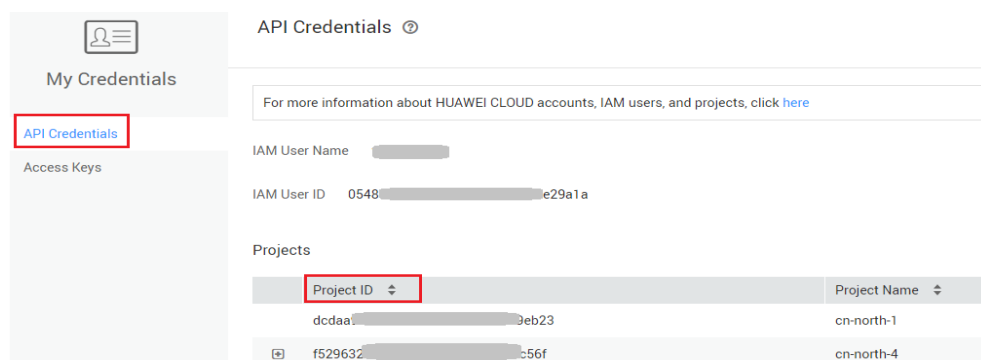
A project ID is required for some URLs when an API is called. You can obtain a project ID on the console.

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

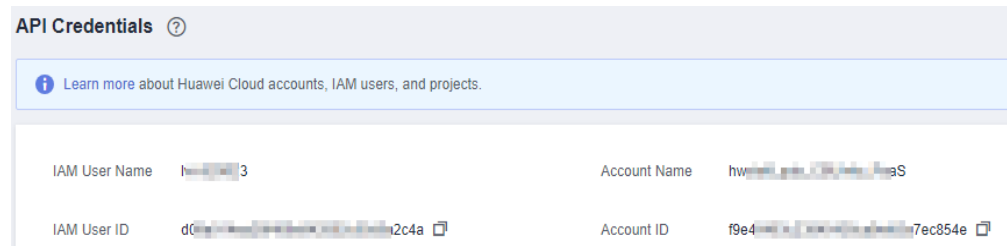
## 8.5 Obtaining the Account Name and Account ID

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 management 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
2023-05-08	Modified the following content: <ul style="list-style-type: none"><li>Added the <b>bss_param</b> parameter for creating a yearly/monthly instance in <a href="#">Creating an Instance</a>.</li></ul>
2022-03-23	Modified the following content: <ul style="list-style-type: none"><li>Modified the APIs for querying product information for instance specification modification and modifying instance specifications to adapt to modification of instances with new flavors.</li></ul>
2021-12-14	Modified the following content: <ul style="list-style-type: none"><li>Changed the APIs from V1 to V2 in <a href="#">Permissions and Supported Actions</a>.</li></ul>
2021-11-16	Modified the following content: <ul style="list-style-type: none"><li>Added V2 APIs.</li></ul>
2020-10-27	This issue is the first official release.