

Alibaba Cloud

Apsara Stack Enterprise

Alibaba Cloud Message Queue Developer Guide

Product Version: v3.16.2

Document Version: 20220815

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings > Network > Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. Getting started	12
1.1. Usage notes on development documentation	12
1.2. Call an API operation	12
2. Preparations	14
2.1. Log on to the API Tool console	14
2.2. View the information about API operations	15
2.3. Obtain an AccessKey pair	15
2.4. Obtain the values of common headers	17
2.5. Obtain the endpoint of Message Queue for Apache Rocket...	18
3. API reference	20
3.1. 文档标题缺失，请补全后重新导出	20
3.1.1. Updates	20
3.1.2. Overview	22
3.1.2.1. Intended audience and scenarios	22
3.1.2.2. Scenarios and limits	22
3.1.2.3. Basic principles	22
3.1.3. Common parameters	22
3.1.3.1. Common request parameters	22
3.1.3.2. Common response parameters	24
3.1.4. Common terms	25
3.1.4.1. Account security	25
3.1.4.1.1. AccessKey	25
3.1.4.1.2. SecretKey	25
3.1.4.2. Resource management	25
3.1.4.2.1. Topic	25
3.1.4.2.2. Group ID	25

3.1.5. Error codes	25
3.1.6. API reference	27
3.1.6.1. Region management	27
3.1.6.1.1. /json/region/list	27
3.1.6.2. Cluster management	28
3.1.6.2.1. /json/cluster/list	28
3.1.6.3. Instance management	29
3.1.6.3.1. /json/vpcTunnel/tunnelSwitch	29
3.1.6.3.2. /json/instance/baseInfo	30
3.1.6.3.3. /json/instance/create	32
3.1.6.3.4. /json/instance/delete	34
3.1.6.3.5. /json/instance/list	34
3.1.6.3.6. /json/instance/update	35
3.1.6.4. Topic management	36
3.1.6.4.1. /json/topic/create	36
3.1.6.4.2. /json/topic/delete	38
3.1.6.4.3. /json/topic/get	39
3.1.6.4.4. /json/topic/listInPage	39
3.1.6.4.5. /json/topic/update	40
3.1.6.4.6. /json/topic/listNames	41
3.1.6.4.7. /json/topic/subDetail	42
3.1.6.5. Group management	43
3.1.6.5.1. /json/group/create	43
3.1.6.5.2. /json/group/delete	44
3.1.6.5.3. /json/group/list	45
3.1.6.5.4. /json/group/subDetail	46
3.1.6.5.5. /json/group/consumerUpdate	48
3.1.6.6. Consumption management	49

3.1.6.6.1. /json/consumer/accumulate	49
3.1.6.6.2. /json/consumer/status	50
3.1.6.6.3. /json/consumer/getConnection	55
3.1.6.6.4. /json/consumer/timespan	57
3.1.6.6.5. /json/consumer/resetoffset	58
3.1.6.7. Message query	59
3.1.6.7.1. /json/message/trace	59
3.1.6.7.2. /json/message/getById	61
3.1.6.7.3. /json/message/getByKey	62
3.1.6.7.4. /json/message/getByPagedTopic	64
3.1.6.8. Publishing and subscription statistics	66
3.1.6.8.1. /json/trend/topicInputTps	66
3.1.6.8.2. /json/trend/groupOutputTps	68
3.1.6.9. Authorization management	70
3.1.6.9.1. /json/empower/create	70
3.1.6.9.2. /json/empower/createBySub	71
3.1.6.9.3. /json/empower/delete	72
3.1.6.9.4. /json/empower/list	72
3.1.6.10. Dead-letter queue management	73
3.1.6.10.1. /json/message/getByIdInDLQ	73
3.1.6.10.2. /json/message/getByPagedDLQ	75
3.1.6.10.3. /json/message/resendByIdInDLQ	77
4.SDK reference	79
4.1. 文档标题缺失，请补全后重新导出	79
4.1.1. Overview	79
4.1.2. Messaging model	79
4.1.3. Updates	79
4.1.4. Transport protocol and character encoding	81

4.1.5. Common classes and methods	82
4.1.5.1. PropertyKeyConst	82
4.1.5.2. ONSFactory	83
4.1.5.2.1. createConsumer	83
4.1.5.2.2. createBatchConsumer	84
4.1.5.2.3. createOrderedConsumer	85
4.1.5.2.4. createOrderProducer	85
4.1.5.2.5. createProducer	86
4.1.5.2.6. createTransactionProducer	87
4.1.5.3. Message	88
4.1.5.3.1. Overview	88
4.1.5.3.2. Default message construction	88
4.1.5.3.3. Message construction by using parameters (1)	88
4.1.5.3.4. Message construction by using parameters (2)	89
4.1.5.3.5. putUserProperties	89
4.1.5.3.6. getUserProperties	89
4.1.5.3.7. getTopic	90
4.1.5.3.8. setTopic	90
4.1.5.3.9. getTag	90
4.1.5.3.10. setTag	91
4.1.5.3.11. getKey	91
4.1.5.3.12. setKey	91
4.1.5.3.13. getMsgID	91
4.1.5.3.14. setMsgID	92
4.1.5.3.15. getReconsumeTimes	92
4.1.5.3.16. setReconsumeTimes	92
4.1.5.3.17. getBornTimestamp	92
4.1.5.3.18. setBornTimestamp	93

4.1.5.3.19. getBornHost	93
4.1.5.3.20. setBornHost	93
4.1.5.3.21. getStartDeliverTime	93
4.1.5.3.22. setStartDeliverTime	94
4.1.5.3.23. getShardingKey	94
4.1.5.3.24. setShardingKey	94
4.1.5.4. ONSClientException	94
4.1.5.4.1. Default ONSClientException construction	94
4.1.5.4.2. ONSClientException construction by using param... ..	95
4.1.5.4.3. ONSClientException construction by using param... ..	95
4.1.5.4.4. ONSClientException construction by using param... ..	95
4.1.5.5. Message sending	95
4.1.5.5.1. Overview	96
4.1.5.5.2. SendResult	96
4.1.5.5.2.1. Default SendResult construction	96
4.1.5.5.2.2. getMessageId	96
4.1.5.5.2.3. getTopic	96
4.1.5.5.2.4. toString	96
4.1.5.5.3. Producer	97
4.1.5.5.3.1. Overview	97
4.1.5.5.3.2. send	97
4.1.5.5.3.3. sendOneway	97
4.1.5.5.3.4. sendAsync	98
4.1.5.5.3.5. isStarted	98
4.1.5.5.3.6. isClosed	98
4.1.5.5.3.7. start	98
4.1.5.5.3.8. shutdown	99
4.1.5.5.4. TransactionProducer	99

4.1.5.5.4.1. send	99
4.1.5.5.4.2. isStarted	99
4.1.5.5.4.3. isClosed	100
4.1.5.5.4.4. start	100
4.1.5.5.4.5. shutdown	100
4.1.5.5.4.6. LocalTransactionExecuter	100
4.1.5.5.4.7. TransactionStatus	101
4.1.5.5.5. OrderProducer	101
4.1.5.5.5.1. send	101
4.1.5.5.5.2. isStarted	102
4.1.5.5.5.3. isClosed	102
4.1.5.5.5.4. start	102
4.1.5.5.5.5. shutdown	102
4.1.5.6. Message subscription	102
4.1.5.6.1. Overview	103
4.1.5.6.2. Consumer	103
4.1.5.6.2.1. Overview	103
4.1.5.6.2.2. subscribe	103
4.1.5.6.2.3. unsubscribe	104
4.1.5.6.2.4. isStarted	104
4.1.5.6.2.5. isClosed	104
4.1.5.6.2.6. start	104
4.1.5.6.2.7. shutdown	105
4.1.5.6.3. MessageListener	105
4.1.5.6.3.1. Overview	105
4.1.5.6.3.2. consume	105
4.1.5.6.4. Action	105
4.1.5.6.4.1. Overview	105

4.1.5.6.4.2. valueOf	106
4.1.5.6.4.3. values	106
4.1.5.6.5. PropertyValueConst	107
4.1.5.6.6. OrderConsumer	107
4.1.5.6.6.1. Overview	107
4.1.5.6.6.2. subscribe	107
4.1.5.6.6.3. unsubscribe	107
4.1.5.6.6.4. isStarted	108
4.1.5.6.6.5. isClosed	108
4.1.5.6.6.6. start	108
4.1.5.6.6.7. shutdown	108
4.1.5.6.7. MessageOrderListener	109
4.1.5.6.7.1. Overview	109
4.1.5.6.7.2. consume	109
4.1.5.6.8. OrderAction	109
4.1.5.6.8.1. Overview	109
4.1.5.6.8.2. valueOf	110
4.1.5.6.8.3. values	110
4.1.5.6.9. BatchConsumer	110
4.1.5.6.9.1. Overview	110
4.1.5.6.9.2. subscribe	110
4.1.5.6.9.3. unsubscribe	111
4.1.5.6.9.4. isStarted	111
4.1.5.6.9.5. isClosed	111
4.1.5.6.9.6. start	112
4.1.5.6.9.7. shutdown	112
4.1.5.6.10. BatchMessageListener	112
4.1.5.6.10.1. Overview	112

4.1.5.6.10.2. consume	112
-----------------------	-----

1. Getting started

1.1. Usage notes on development documentation

This topic describes how to view the desired product development documentation when you use Apsara Stack API (ASAPI) in your development workflow.

Prerequisites

The Apsara Stack version is obtained from an O&M engineer or an operations administrator.

Procedure

You can go to the Apsara Stack documentation center and view the desired product development documentation.

1. Go to the [Apsara Stack Documentation Center](#) page by using a browser.
2. Click the **Enterprise**, **Agility**, or **All-in-one** tab based on the Apsara Stack version that you use.
3. Select the Apsara Stack version from the drop-down list.

4. In the left-side navigation pane, select the name of the product that you want to view.
5. On the right side of the product documentation section, click **View Details** or **Download PDF**.

1.2. Call an API operation

You can use the point of presence (POP) gateway or the Apsara Stack API (ASAPI) gateway to call API operations in Apsara Stack. In Apsara Stack Enterprise V3.16.0 and later, the POP gateway is used by default to ensure that the user experience is consistent with that in the Alibaba Cloud public cloud. This topic describes how to use the POP gateway to call an API operation in Apsara Stack.

Context

You can use the POP gateway or the ASAPI gateway to call API operations in Apsara Stack. In Apsara Stack Enterprise V3.16.0 and later, the POP gateway is used by default to ensure that the user experience is consistent with that in the Alibaba Cloud public cloud. This topic describes how to use the POP gateway to call an API operation in Apsara Stack.

 **Note** You can continue to use the ASAPI gateway to call API operations in Apsara Stack whose version is earlier than V3.16.0. For more information, see [Call the ASAPI gateway](#), [Obtain an endpoint of ASAPI](#), and [Obtain Apsara Stack SDKs](#).

Procedure

1. Obtain the AccessKey pair.
Obtain the AccessKey pair that is required for the API call. For more information, see [Obtain an AccessKey pair](#).

 **Note** We recommend that you use the AccessKey pair of a personal account to call the API operations of the Apsara Uni-manager Operations Console and cloud services.

2. Obtain the region ID.

Obtain the region ID that is required for the API call. For more information, see [Obtain the values of common headers](#).

3. Obtain the organization ID and the resource set ID.

You can call API operations to obtain the organization ID and the resource set ID.

- For more information about how to obtain an organization ID, see the *GetOrganizationList* section in *Apsara Uni-manager Operations Console Developer Guide*.
- For more information about how to obtain a resource set ID, see the *ListResourceGroup* section in *Apsara Uni-manager Operations Console Developer Guide*.

 **Note** When you call an API operation, use the `<x-acs-organizationid>` and `<x-acs-resourcegroupid>` headers to specify the organization and resource set to which the instance resource belongs. This way, you can query instance resources in the specified organization and resource set.

- If you do not configure `<x-acs-organizationid>`, the organization to which the current user belongs is used by default.
- If you do not configure `<x-acs-resourcegroupid>`, this header is empty by default. If you specify a resource set ID, you must also specify an organization ID.

4. View the developer guide of the cloud service that you want to use to learn how to make the API call.

Obtain the developer guide of the cloud service that you want to use. This way, you can learn the features of API operations and how to call the API operations. For information about how to obtain a developer guide, see [Usage notes on development documentation](#).

5. Obtain the endpoint that is used to access Message Queue for Apache RocketMQ.

6. Call the API.

2.Preparations

2.1. Log on to the API Tool console

You can log on to the API Tool console from the Apsara Uni-manager Management Console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment personnel.
- A browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press Enter.
2. Enter your username and password.

Obtain the username and password that are used to log on to the Apsara Uni-manager Management Console from the operations administrator.

 **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your account. For security purposes, your password must meet the minimum complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, which include ! @ # \$ %

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click **Log On**.
 - c. Enter a six-digit MFA verification code and click **Authenticate**.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click **Authenticate**.

 **Note** For more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Others > API Tool**.

2.2. View the information about API operations

This topic describes how to view the information about API operations. The information includes service names, API operation names, and API versions. The information is required when you use an SDK to call API operations.

Context

The API operations on the API Directory page are only for reference. For information about how to call API operations, see developer guides of the corresponding cloud services. If the information on the API Directory page is inconsistent with that in the developer guide of the corresponding cloud service, the information in the developer guide prevails.

Procedure

1. Go to the API Tool console. For more information, see [Log on to the API Tool console](#).
2. In the left-side navigation pane, click **API Directory**.
3. On the **API Directory** page, select the service that you want to use from the drop-down list, enter a keyword of an API operation name in the search box, and then click Search.

API Directory		
All Products ▾	cre	Q
Product Name	Operation Name	API Version
cr	CreateMetadataOccurrence	2018-12-01
cr	CreateUserInfo	2016-06-07
cr	CreateNamespaceAuthorization	2016-06-07
cr	CreateNamespace	2016-06-07

4. View and record the service name, API operation name, and API version of the API operation that you want to call.

2.3. Obtain an AccessKey pair

AccessKey pairs are classified into AccessKey pairs assigned by Resource Access Management (RAM) and those assigned by Security Token Service (STS). You can use either type of AccessKey pair for authentication when you call operations. This topic describes how to obtain an AccessKey pair that is assigned by RAM.

Context

An AccessKey pair is used to verify the identity of a request sender. An AccessKey pair consists of an AccessKey ID and an AccessKey secret that are used to implement symmetric encryption. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt a signature string.

You can use RAM or STS to assign an AccessKey pair for authentication.

- Use RAM to assign an AccessKey ID and an AccessKey secret to a third party.
- Use STS to assign an AccessKey ID, an AccessKey secret, and a security token to a third party. For more information, see [Obtain an AccessKey pair assigned by STS](#).

The Apsara Uni-manager Management Console provides AccessKey pairs for personal accounts and organizations. Only operations administrators and the administrators of level-1 organizations can obtain the AccessKey pairs of organizations. We recommend that you use the AccessKey pairs of personal accounts to call the operations provided by the Apsara Uni-manager Management Console and cloud services. If you use the AccessKey pair of a personal account, configure the request headers that are described in the following table for access control. Otherwise, an error that indicates insufficient permissions may occur.

 **Warning** The AccessKey pair of a personal account is managed by the permission system of the Apsara Uni-manager Management Console. If you need to obtain the AccessKey pair of an organization, make sure that your operation is allowed by an administrator. This ensures security.

Parameter	Description
x-acs-regionid	The ID of the region. Example: cn-hangzhou-*
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console. For more information about how to obtain organization IDs, see the <i>GetOrganizationList</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console. For more information about how to obtain resource set IDs, see the <i>ListResourceGroup</i> section in <i>Apsara Uni-manager Management Console Developer Guide</i> .
x-acs-instanceid	The ID of the instance that you want to manage. If you want to obtain the ID of an Elastic Compute Service (ECS) instance, you can obtain the ID from the ECS instance list.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console.
2. In the upper-right corner of the homepage, click the profile picture and then click **User Information**.
3. In the **AccessKey Pair** section, view the AccessKey pair of your account.

AccessKey					Create AccessKey Pair
 The Accesskey ID and AccessKey Secret are the keys used to access the cloud service resources. They have full permissions on the account. Please keep them properly.					
AccessKey ID	AccessKey Secret	Status	Created At	Actions	
W1YRFvKH...		 Enable	Sep 18, 2021, 11:05:57	Disable	Delete
IsJsAUyX...		 Enable	Sep 10, 2021, 12:28:35	Disable	Delete

 **Note** An AccessKey pair consists of an AccessKey ID and an AccessKey secret. These credentials provide full permissions on cloud resources within your account. You must keep the AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following steps:

1. Log on to the Apsara Uni-manager Management Console as an administrator.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Organizations**.
4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
5. Click **Management Accesskey**.
6. In the dialog box that appears, view the AccessKey pair of the organization.

2.4. Obtain the values of common headers

You must provide multiple headers when you call API operations in Apsara Stack. This topic describes the headers and how to obtain the values of the headers. This topic also describes the return values of specific headers after you call API operations in Apsara Stack.

Headers

Header	Description	Method to obtain
x-acs-regionid	The ID of the region.	Obtain a region ID
x-acs-organizationid	The ID of the organization in the .	Obtain an organization ID
x-acs-resourcegroupid	The ID of the resource set in the .	Obtain a resource set ID
x-acs-instanceid	The ID of the Message Queue for Apache RocketMQ instance on which you want to perform operations.	Obtain an instance ID

Obtain a region ID

1. Log on to the .
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane, choose **Resources > Regions**.
4. Click the name of the organization that you want to manage.
5. View the region ID in the **Regions** section.

Obtain an organization ID

You can call the `GetOrganizationList` operation to obtain an organization ID. For more information, see the *GetOrganizationList* section in *Developer Guide*.

Obtain a resource set ID

You can call the `ListResourceGroup` operation to obtain a resource set ID. For more information, see the *ListResourceGroup* section in *Developer Guide*.

Obtain an instance ID

1. Log on to the .
2. In the top navigation bar, choose **Products > Middleware > Message Queue**.
3. In the left-side navigation pane, click **Instances**.
4. In the top navigation bar of the **Instances** page, click the name of the instance that you want to view. On the **Instance Information** tab, view the instance ID.

Response parameters

The following additional response parameters are returned by default after you call API operations to query the instances of cloud services.

Parameter	Description
ResourceGroupName	The name of the resource set.
ResourceGroup	The ID of the resource set.
DepartmentName	The name of the organization.
Department	The ID of the organization.

2.5. Obtain the endpoint of Message Queue for Apache RocketMQ

This topic describes how to obtain the endpoint of . To use the POP gateway to call the API operations of an Apsara Stack service, you must specify the endpoint of the Apsara Stack service. You can contact an onsite O&M engineer of Apsara Stack to obtain a service endpoint. You can also use the method provided in this topic to obtain a service endpoint.

Procedure

1. Log on to the Apsara Infrastructure Management console.
 - i. Log on to the Apsara Uni-manager Operations Console.
 - ii. In the top navigation bar, choose **Products > Platforms > Apsara Infrastructure Management Framework**.
2. In the left-side navigation pane, click **Reports**.
3. On the **All Reports** page, search for *Registration Vars of Services*. In the report list that appears, click the name of the report.

4. On the **Registration Vars of Services** page, click  next to **Service** and search for *mq*.
5. Find the **mq-console** service, right-click the **Service Registration** column, and then select **Show More** from the shortcut menu.

On the **Details** page, the value of the **ons_innerapi_endpoint** parameter indicates the endpoint of .

3.API reference

3.1. 文档标题缺失，请补全后重新导出

3.1.1. Updates

This topic describes the updates of Message Queue for Apache RocketMQ from V3.8.0 to V3.8.1 to help you get started with the updated version.

Optimization of resource isolation by instance

Message Queue for Apache RocketMQ provides instances for multi-tenancy isolation. Each user can purchase multiple instances and logically isolate them from each other.

To ensure the compatibility with the existing resources of existing users, Message Queue for Apache RocketMQ provides the following types of instances and namespaces:

- Default instances, which are compatible with the existing resources of existing users
 - This type of instance has no separate namespace. Resource names must be globally unique within and across all instances.
 - By default, an instance without a namespace is automatically generated for the existing resources of each existing user. If no existing resources are available, you can create at most one instance without a namespace.
 - You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxxx");
// Compatible configuration, which is not recommended. We recommend that you update this
// configuration to the recommended configuration:
properties.put(PropertyKeyConst.ONSAddr, "xxxx");
```

- New instances
 - A new instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances.
 - You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
```

- A RocketMQ client must be updated to the following latest versions for different programming languages:
 - Java: **V1.8.7.1.Final**
 - C and C++: **V2.0.0**
 - .NET: **V1.1.3**

Optimization of resource application

Previously, Message Queue for Apache RocketMQ resources consisted of topics, producer IDs, and consumer IDs. Each two of the resources have a many-to-many relationship, which was difficult to comprehend. Each time you created a topic, you must associate the topic with a producer ID and a consumer ID. This process was too complex for medium- and large-sized enterprise customers.

To optimize user experience and help new users get started, the resource application process has been simplified.

The resource application process has been optimized in the following aspects:

- Topic management, which is unchanged
 - You need to apply for a topic. A topic is used to classify messages. It is the primary classifier.
- Group management
 - You do not need to apply for a producer ID. Producer IDs and consumer IDs are integrated into group IDs. In the Message Queue for Apache RocketMQ console, the Producers module has been removed. The Producers and Consumers modules have been integrated into the Groups module.
 - You do not need to associate a producer ID or consumer ID with a topic. Instead, you need only to apply for a group ID and associate it with a topic in the code.
 - Compatibility:
 - The list of producer IDs is no longer displayed. This does not affect the current services.
 - The consumer IDs that start with CID- or CID_ and that you have applied for can still be used and can be set in the `PropertyKeyConst.ConsumerId` or `PropertyKeyConst.GROUP_ID` parameter of the code.
- Sample code

Note

- We recommend that you update a RocketMQ client to the following latest versions for different programming languages:
 - Java: **V1.8.7.1.Final**
 - C and C++: **V2.0.0**
 - .NET: **V1.1.3**
- Existing producer IDs or consumer IDs can still be used and do not affect the current services. However, we recommend that you update your instance configuration to the recommended configuration.

- Recommended configuration: Integrate producer IDs and consumer IDs into group IDs.

```
// Set the PropertyKeyConst.GROUP_ID parameter. The original PropertyKeyConst.ProducerId and PropertyKeyConst.ConsumerId parameters are deprecated.
properties.put(PropertyKeyConst.GROUP_ID, "The original CID-XXX or the GID-XXX");
```

- Compatible configuration: Use a producer ID to identify a producer and a consumer ID to identify a consumer.

```
// When you create a producer, you must set the PropertyKeyConst.ProducerId parameter.
properties.put(PropertyKeyConst.ProducerId, "The original PID-XXX or the GID-XXX");
// When you create a consumer, you must set the PropertyKeyConst.ConsumerId parameter.
properties.put(PropertyKeyConst.ConsumerId, "The original CID-XXX or the GID-XXX");
```

3.1.2. Overview

3.1.2.1. Intended audience and scenarios

Message Queue for Apache RocketMQ provides you with API operations to help you manage resources. Different from the SDKs for production, API operations are used to manage the resources of Message Queue for Apache RocketMQ, such as topics and group IDs.

API operations are intended for those who:

- Are familiar with the main features of the Message Queue for Apache RocketMQ console.
- Require a management and control tool similar to the console and have a need for custom management and control features.

3.1.2.2. Scenarios and limits

1. Due to the implementation principles of API operations, the caller must be able to access the domain name of the Message Queue for Apache RocketMQ console. Otherwise, the call may fail.
2. The underlying layer of API operations is based on HTTP, with average performance and efficiency. Therefore, API operations are applicable only to management and control scenarios. Do not use them in mass production scenarios with high concurrency.

3.1.2.3. Basic principles

The underlying layer of API operations provides services based on HTTP. Each time you call an API operation, you send an HTTP request from the HTTP client to the backend service of Message Queue for Apache RocketMQ for execution.

You can encapsulate parameters in an HTTP request and send the request to call an API operation. The execution results are returned in a response. The request and response are encoded in UTF-8.

The service endpoint in the HTTP request is the domain name of the Message Queue for Apache RocketMQ console.

3.1.3. Common parameters

3.1.3.1. Common request parameters

The HTTP client calls Message Queue for Apache RocketMQ API operations based on HTTPS. Common request parameters must be included in all API requests and sent to the backend service of Message Queue for Apache RocketMQ for permission verification.

The following table describes common request parameters.

Parameter	Type	Required	Description
_accesskey	String	Yes	The AccessKey ID of your account.

Parameter	Type	Required	Description
__preventCache	Long	Yes	The parameter that is used for cross-site request forgery (CSRF) verification. Set this parameter to the current system time. Unit: milliseconds.
_regionId	String	Yes (except for the operation that is used to query region IDs)	The ID of the current region.
_signature	String	Yes	The signature string of the request.

The `_regionId` parameter must specify the region where you want to manage resources. The default value is `mq-private` for private domains and Apsara Stack.

The `_signature` parameter specifies the signature string that is obtained from signature calculation. During signature calculation, other parameters in the HTTPS request are arranged to create a string-to-sign. Then, the string-to-sign is calculated and signed based on the AccessKey secret of your account.

Signature calculation example

Assume that the original HTTPS request is:

```
https://mq.console.${global:internet-domain}/json/topic/create
```

 **Note** To access the region in single-region deployment or access the central region in multi-region deployment, use `${global:internet-domain}` to specify the domain name to be accessed. To access a regular region in multi-region deployment, use `${global:region}.${global:internet-domain}` to specify the domain name to be accessed.

The request contains the following parameters, including common parameters and operation-specific parameters:

`Key_2=Value_2&Key_1=Value_1&Key_3=Value_3`

Then, the original request URL is:

```
https://mq.console.${global:internet-domain}/json/topic/create?Key_2=Value_2&Key_1=Value_1&Key_3=Value_3
```

Sort the keys `Key_2`, `Key_1`, and `Key_3` in alphabetical order and connect the corresponding values in the same order to form a string. The sorting result of the keys is `Key_1`, `Key_2`, `Key_3` and the value string is `Value1Value2Value3`. Then, the string-to-sign is `Value1Value2Value3`.

Use the AccessKey secret of your account to sign the string-to-sign by using the HMAC-SHA1 algorithm. Then, the signature string is obtained. Function libraries are available for the implementation of the HMAC algorithm in different programming languages. You can search for one as required.

For example, the following method can be used in Linux:


```
echo -n 'Value1Value2Value3' | openssl dgst -hmac 'Your AccessKey secret' -sha1 -binary | base64
```

Assume that the final result is: eqweq+adwe23****=.

Replace "+" with "%2B", "=" with "%3D", and "/" with "%2F" by using curl commands. Otherwise, an authentication failure is reported after the characters are automatically escaped. The signed HTTPS request is:

```
https://mq.console.${global:internet-domain}/json/topic/create?Key_2=Value_2&Key_1=Value_1&Key_3=Value_3&_signature=eqweq%2Badwe23****%3D
```

3.1.3.2. Common response parameters

The HTTP client sends an HTTP request to the backend service of Message Queue for Apache RocketMQ. The backend service processes the request and then returns results in an HTTP response to the client in the same way. In normal cases, the HTTP status code is 200 in the response. The returned content is contained in the body as a JSON string. The client needs to parse the JSON string to determine whether the call is successful.

JSON format:

```
{
  "code": 200,
  "currentPage": 0,
  "data": { },
  "dynamicCode": "",
  "dynamicMessage": "",
  "message": "operation success",
  "pageSize": 0,
  "requestId": "",
  "status": "SUCCESS",
  "success": true,
  "total": 0
}
```

Response parameters:

- **code**: the HTTP status code. A value of 200 indicates that the call is successful. Other values indicate that the call fails.
- **currentPage**: the page number of the returned page if the request is a paged query.
- **dynamicMessage**: the error message.
- **message**: the processing result.
- **pageSize**: the number of entries returned per page if the request is a paged query.
- **success**: indicates whether the call is successful.
- **data**: the returned service data. The value is also a JSON string. If the return results of the current request contain service data, you must reverse parse the content in the JSON format.
- **RequestId**: the unique ID of the request. The value is a string.

3.1.4. Common terms

3.1.4.1. Account security

3.1.4.1.1. AccessKey

The AccessKey ID of the account that you use to access Alibaba Cloud services. You can obtain your AccessKey ID from the console. When you call an API operation, the AccessKey ID is included in the API request and sent to the backend service of Message Queue for Apache RocketMQ for identity authentication. You can call Message Queue for Apache RocketMQ API operations to manage resources only within your account.

3.1.4.1.2. SecretKey

The AccessKey secret of the account that you use to access Alibaba Cloud services. You can obtain your AccessKey secret from the console. When you call an API operation, the AccessKey secret is included in the API request and used to calculate the signature string of the API request. The backend service of Message Queue for Apache RocketMQ uses the AccessKey secret to verify the signature string of the API request for authentication.

3.1.4.2. Resource management

3.1.4.2.1. Topic

A classifier that is used to classify messages. Messages in Message Queue for Apache RocketMQ are stored by topic. Topics are among the most important resources in Message Queue for Apache RocketMQ. In Message Queue for Apache RocketMQ, topics are globally unique.

3.1.4.2.2. Group ID

The ID of a group. A group consists of producers or consumers that produce or consume messages of the same type and publish or subscribe to messages based on the same logic.

- If group IDs for consumers or existing consumer IDs are created by a RAM user, both the RAM user and the Apsara Stack tenant account to which the RAM user belongs have permissions on these resources.
- If group IDs for consumers or existing consumer IDs are created by an Apsara Stack tenant account, only the Apsara Stack tenant account has permissions on these resources. Its RAM users have no permissions on these resources.
- For more information about the changes in the use of existing producer IDs, existing consumer IDs, and group IDs, see [Updates](#).

For more information about the changes in the use of existing producer IDs, existing consumer IDs, and group IDs, see [Updates](#).

3.1.5. Error codes

If a Message Queue for Apache RocketMQ API operation is successfully called, HTTP status code 200 is returned to the client. If the call fails, error codes and descriptions are returned. You can find solutions based on the error codes that are described in the following table.

Error code	Problem description and cause	Solution
ONS_SYSTEM_ERROR	An error has occurred at the backend of Message Queue for Apache RocketMQ.	Submit a ticket to contact the technical support staff of Message Queue for Apache RocketMQ.
ONS_SERVICE_UNSUPPORTED	The API operation is not supported in the specified region.	Contact the technical support staff of Message Queue for Apache RocketMQ to confirm whether the operation is supported.
ONS_INVOKE_ERROR	The API operation failed to be called.	Contact the technical support staff of Message Queue for Apache RocketMQ.
BIZ_FIELD_CHECK_INVALID	Parameters failed the verification.	Check whether the specified request parameters are valid based on the API reference.
BIZ_TOPIC_NOT_FOUND	The topic does not exist.	Check whether the specified topic is valid or whether the topic is created.
BIZ_SUBSCRIPTION_NOT_FOUND	The group ID of the subscription does not exist.	Check whether the group ID is created or whether the query conditions are valid.
BIZ_PUBLISHER_EXISTED	The specified group ID already exists.	Use a new group ID and try again.
BIZ_SUBSCRIPTION_EXISTED	The specified group ID already exists.	Use a new group ID and try again.
BIZ_CONSUMER_NOT_ONLINE	The consumer that is identified by the specified group ID is offline.	Ensure that the consumer is online and try again.
BIZ_NO_MESSAGE	No messages match the query conditions.	Check the query conditions and check whether messages are sent based on the query conditions.

Error code	Problem description and cause	Solution
BIZ_REGION_NOT_FOUND	The requested region does not exist.	Check whether the specified region ID is valid.
BIZ_TOPIC_EXISTED	The specified topic already exists.	Use a new topic name and try again.
BIZ_PRODUCER_ID_BELONG_TO_OTHER_USER	The specified group ID is occupied by another user.	Use a new group ID and try again.
BIZ_CONSUMER_ID_BELONG_TO_OTHER_USER	The specified group ID is occupied by another user.	Use a new group ID and try again.
BIZ_PUBLISH_INFO_NOT_FOUND	The requested group ID does not exist.	Check whether the group ID exists or whether the query conditions are valid.
EMPOWER_EXIST_ERROR	The specified authorization already exists.	Check the request parameters and try again or query the authorization first.
EMPOWER_OWNER_CHECK_ERROR	The current user is not the owner of the authorized topic.	Check the ownership of the resource.

3.1.6. API reference

3.1.6.1. Region management

3.1.6.1.1. /json/region/list

Queries region IDs. The queried region IDs are used as the value of the common request parameter `_regionId` for other operations.

Scenarios

You can call this operation to query region IDs. The queried region IDs are used as the value of the common request parameter `_regionId` for other operations.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List<RegionDo>	The query results.

RegionDo

Parameter	Type	Description
id	Long	The number in the database.
regionId	String	The ID of the region. Example: cn-hangzhou.
regionName	String	The name of the region. Example: China (Hangzhou).

3.1.6.2. Cluster management

3.1.6.2.1. /json/cluster/list

Queries clusters. When you create an instance, you must use a cluster name to specify the relationship between the instance and the cluster. You can call this operation to query cluster names.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
_regionId	String	Yes	The ID of the region where you want to query clusters.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List<ClusterInfoDo>	The query results.

ClusterInfoDo

Parameter	Type	Description
regionId	String	The ID of the region where the cluster resides.
cluster	String	The name of the cluster.
createTime	String	The time when the cluster was created.
modifyTime	String	The last time when the cluster was modified.

3.1.6.3. Instance management

3.1.6.3.1. /json/vpcTunnel/tunnelSwitch

Switches the network access mode for virtual private clouds (VPCs) in a cluster and applies for a virtual IP address (VIP) in the specified access mode for RocketMQ brokers and RocketMQ Name Servers in the cluster.

Scenarios

An instance provides two network access modes for VPCs: Single Tunnel and Any Tunnel.

Important notes:

- The network access mode for VPCs is switched in a specified cluster. This affects all topics in which messages are being sent and received in the cluster. Therefore, check the impact scope before you call this operation.
- After the network access mode for VPCs is switched in a cluster, change the configuration of all clients that are connected to the cluster to the return value of this operation:
`properties.put(PropertyKeyConst.NAMESRV_ADDR,"Return value");`

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
tunnelType	String	Yes	The network access mode for VPCs. Valid values: anyTunnel and singleTunnel.
instanceId	String	Yes	The ID of the instance whose network access mode for VPCs you want to switch.
slbUrl	String	No	The endpoint of the Server Load Balancer (SLB) API.

Parameter	Type	Required	Description
force	Boolean	No	Specifies whether to forcibly switch the network access mode for VPCs. A value of true indicates that the network access mode for VPCs is forcibly switched regardless of whether RocketMQ brokers are processing messages in the cluster. A value of false indicates that the switching is blocked if the sum of the transactions per second (TPS) for receiving messages and the TPS for sending messages is greater than 10 on a single RocketMQ broker. Default value: false.
vSwitchId	String	No	The VPC in which the VIP in Single Tunnel mode is created. This parameter takes effect only when the tunnelType parameter is set to singleTunnel. If you have only one VPC or the tunnelType parameter is set to anyTunnel, this parameter can be left unspecified.

Response parameters

The VIP of RocketMQ Name Servers is returned in the JSON format. You must change the configuration of all clients that are connected to the cluster to `properties.put(PropertyKeyConst.NAMESRV_ADDR,"Return value");`. Then, restart your application to make the configuration take effect.

3.1.6.3.2. /json/instance/baseInfo

Queries the basic information about an instance, including the endpoints for sending and receiving messages.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
InstanceBaseInfo	InstanceBaseInfo	The information about the instance.

InstanceBaseInfo

Parameter	Type	Description
instanceId	String	The ID of the instance.
instanceName	String	The name of the instance.
independentNaming	Boolean	Indicates whether the instance has a namespace. Valid values: • true: The instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances. • false: The instance has no separate namespace. Resource names must be globally unique within and across all instances. You can view the information about the namespace on the Instance Details page in the Message Queue for Apache RocketMQ console. For more information about namespaces, see Updates.
Endpoints	Endpoints	The endpoints for various protocols.
maxSendTps	Long	The maximum TPS for sending messages in the instance.
maxReceiveTps	Long	The maximum TPS for receiving messages in the instance.

Parameter	Type	Description
topicCapacity	Integer	The maximum number of topics that can be created in the instance.

Endpoints

Parameter	Type	Required	Description
tcpEndpoint	String	No	The Message Queue for Apache RocketMQ endpoint for TCP.

3.1.6.3.3. /json/instance/create

Creates an instance.

Note

- By default, you must use your Apsara Stack tenant account to call this operation. RAM users can call this operation only after they are authorized.
- A region can contain up to 500 instances.
- The name of an instance must be unique in the region where the instance is deployed.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
cluster	String	Yes	The cluster in which you want to create the instance.
instanceName	String	Yes	The name of the instance. The name must be 3 to 64 characters in length.

Parameter	Type	Required	Description
independentNaming	Boolean	Yes	Specifies whether the instance has a namespace. Valid values: • true: The instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances. • false: The instance has no separate namespace. Resource names must be globally unique within and across all instances. You can view the information about the namespace on the Instance Details page in the Message Queue for Apache RocketMQ console. For more information about namespaces, see Updates.
remark	String	No	The description of the instance. You can leave this parameter unspecified.
maxSendTps	Long	Yes	The maximum TPS for sending messages in the instance.
maxReceiveTps	Long	Yes	The maximum TPS for receiving messages in the instance.
topicCapacity	Integer	No	The maximum number of topics that can be created in the instance.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.3.4. /json/instance/delete

Deletes an instance.

Note

- By default, you must use your Apsara Stack tenant account to call this operation. RAM users can call this operation only after they are authorized.
- An instance can be deleted only when it has no topics or group IDs.

An instance can be deleted only when it has no topics or group IDs.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.3.5. /json/instance/list

Queries all instances in the current environment.

Scenarios

You can call this operation to query the information about all instances for instance management.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List<InstanceVO>	The information about all existing instances.

InstanceVO

Parameter	Type	Description
instanceId	String	The ID of the instance.
instanceName	String	The name of the instance.

Parameter	Type	Description
namespaceRulesType	Boolean	Indicates whether the instance has a namespace. Valid values: • true: The instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances. • false: The instance has no separate namespace. Resource names must be globally unique within and across all instances. You can view the information about the namespace on the Instance Details page in the Message Queue for Apache RocketMQ console. For more information about namespaces, see Updates.

3.1.6.3.6. /json/instance/update

Updates the name and description of an instance.

Note

- By default, you must use your Apsara Stack tenant account to call this operation. RAM users can call this operation only after they are authorized.
- A region can contain up to 500 instances.
- The name of an instance must be unique in the region where the instance is deployed.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.

Parameter	Type	Required	Description
instanceName	String	No	The name of the instance. The name must be 3 to 64 characters in length. If you do not specify this parameter, the name of the instance is not updated.  Note The instance name must be unique in a region.
remark	String	No	The description of the instance. If you do not specify this parameter, the description of the instance is not updated.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.4. Topic management

3.1.6.4.1. /json/topic/create

Creates a topic in a region.

Scenarios

You can call this operation when new topics are required to publish new applications or implement service logic.

By default, you must use your Apsara Stack tenant account to call this operation. RAM users can call this operation only after they are authorized.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
topic	String	Yes	The name of the topic that you want to create.  Note • CID and GID are the reserved fields of a group ID. A topic name cannot start with CID or GID. • If a namespace is configured for the instance where you want to create the topic, the topic name must be unique within the instance but can be the same as topic names in other instances. • If no namespace is configured for the instance where you want to create the topic, the topic name must be globally unique within the instance and across instances and regions.
instanceld	String	Yes	The ID of the instance where the topic you want to create resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
orderType	Integer	Yes	The type of message in the topic.  Note • 0: regular message • 1: partitionally ordered message • 2: globally ordered message • 4: transactional message • 5: scheduled or delayed message
remark	String	No	The description of the topic. You can leave this parameter unspecified.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Examples

Signature calculation method for the `_signature` parameter. For more information, see [Common request parameters](#). Example:

```
echo -n '1498029701000ACCESSKEY_XXXonsConsolemq-privateTOPIC_XXX' | openssl dgst -hmac 'SECRETKEY_XXX' -sha1 -binary | base64
```

Call method:

```
curl http://mq.console.example.com//json/topic/create?_accesskey=ACCESSKEY_XXX&__preventCache=1498029701000&_platform=onsConsole&_regionId=mq-private&topic=TOPIC_XXX&_signature=8DqV%2Br0BzOd0M2SNj3Acuqn***%3D
```

 **Note** Contact an administrator to obtain the project-specific domain name, and replace `example.com` with the actual domain name.

Sample response:

```
{"code":200,"currentPage":0,"data":{},"dynamicCode":"","dynamicMessage":"","helpUrl":null,"message":"operation success","pageSize":0,"requestId":"","status":"SUCCESS","success":true,"total":0}
```

3.1.6.4.2. /json/topic/delete

Deletes a topic within your account.

Scenarios

You can call this operation when you need to recycle resources. For example, when applications are deleted, you need to recycle relevant resources. After a topic is deleted, the backend of Message Queue for Apache RocketMQ recycles the resources of the topic. This process is time-consuming. Therefore, we recommend that you do not create a topic immediately after you delete one. If a topic fails to be deleted, troubleshoot the issue based on the returned error code.

 **Notice** The operation to delete a topic results in the cleanup of all subscriptions under the topic. Proceed with caution.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
topic	String	Yes	The name of the topic that you want to delete.

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the topic you want to delete resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.4.3. /json/topic/get

Queries a topic.

Scenarios

You can call this operation to query the basic information about a topic, such as the owner and status.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
topic	String	Yes	The name of the topic that you want to query.
instanceld	String	Yes	The ID of the instance where the topic you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.4.4. /json/topic/listInPage

Queries one or more topics within your account by page.

Scenarios

You can call this operation to query the information about one or more topics within your account by page. You can call the listNames operation to query the names of all topics, and then call this operation to query the detailed information about topics.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
topic	String	No	The name of the topic that you want to query.
currentPage	Integer	No	The number of the page to return. Default value: 1.
pageSize	Integer	No	The number of topics to return on each page. Default value: 20.
instanceld	String	Yes	The ID of the instance for which you want to query topics.

Response parameters

The specified number of topics on the specified page is returned in the JSON format.

3.1.6.4.5. /json/topic/update

Configures the read/write mode for a topic.

Scenarios

By default, you must use your Apsara Stack tenant account to call this operation. RAM users can call this operation only after they are authorized.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the topic you want to configure resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The name of the topic that you want to configure. A topic name must be unique within your account even across different regions.
perm	Integer	Yes	The read/write mode that you want to configure for the topic. Valid values: • 6: Both read and write operations are supported. • 4: Write operations are prohibited. • 2: Read operations are prohibited.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.4.6. /json/topic/listNames

Queries the names of all topics within your account. You can use this operation together with the listInPage operation to query the detailed information about all topics.

Scenarios

You can call this operation to query the names of all topics within your account.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance for which you want to query the names of topics. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.

Response parameters

The names of all topics that are created within your account are returned in the JSON format.

3.1.6.4.7. /json/topic/subDetail

Queries the online consumer groups that have subscribed to a topic.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the topic you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The name of the topic that you want to query.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The returned data.

Data

Parameter	Type	Description
topic	String	The name of the topic.
SubscriptionDataList	List<SubscriptionDataListItem>	The information about online consumer groups.

SubscriptionDataListItem

Parameter	Type	Description
groupId	String	The ID of the group that has subscribed to the topic.
subString	String	The subscription string.
messageModel	String	The consumption mode. Valid values: - CLUSTERING: clustering consumption - BROADCASTING: broadcasting consumption

3.1.6.5. Group management

3.1.6.5.1. /json/group/create

Creates a group ID.

Usage notes

When you call this operation, take note of the following items:

- If group IDs for consumers or existing consumer IDs are created by a RAM user, both the RAM user and the Apsara Stack tenant account to which the RAM user belongs have permissions on these resources.
- If group IDs for consumers or existing consumer IDs are created by an Apsara Stack tenant account, only the Apsara Stack tenant account has permissions on these resources. Its RAM users have no permissions on these resources.
- For more information about the changes in the use of existing producer IDs, existing consumer IDs, and group IDs, see [Updates](#).

When you call this operation, take note of the changes in the use of existing producer IDs, existing consumer IDs, and group IDs. For more information, see [Updates](#).

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the group ID you want to create resides.
groupid	String	Yes	The group ID that you want to create. Take note of the following items: - The group ID must start with GID_ or GID- and can contain letters, digits, hyphens (-), and underscores (_). - The group ID must be 5 to 64 characters in length.  Note The group ID cannot be modified after it is created.
remark	String	No	The description of the group ID.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
remark	String	The description of the group ID.

3.1.6.5.2. /json/group/delete

Deletes a group ID that is created after you call the /json/group/create operation.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the group ID you want to delete resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
groupId	String	Yes	The group ID that you want to delete.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.5.3. /json/group/list

Queries one or more group IDs.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance for which you want to query group IDs. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.

Parameter	Type	Required	Description
groupId	String	No	The group ID that you want to query. If you do not specify this parameter, all group IDs are queried.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List(SubscribeInfoDo)	The subscription information.

SubscribeInfoDo

Parameter	Type	Description
owner	String	The owner of the group ID.
instanceId	String	The ID of the instance.
groupId	String	The ID of the consumer group.
independentNaming	Boolean	Indicates whether the instance has a namespace. Valid values: • true: The instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances. • false: The instance has no separate namespace. Resource names must be globally unique within and across all instances. You can view the information about the namespace on the Instance Details page in the Message Queue for Apache RocketMQ console. For more information about namespaces, see Updates.
remark	String	The description of the group ID.

3.1.6.5.4. /json/group/subDetail

Queries the topics to which a group has subscribed. No data can be queried if the group is offline.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the group ID you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
groupId	String	Yes	The group ID that you want to query.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The returned data.

Data

Parameter	Type	Description
groupId	String	The group ID that you queried.
online	Boolean	Indicates whether the group was online.
messageModel	String	The consumption mode. Valid values: - CLUSTERING: clustering consumption - BROADCASTING: broadcasting consumption

Parameter	Type	Description
SubscriptionDataList	List<SubscriptionDataListItem>	The information about topics. No data can be queried if the group was offline.

SubscriptionDataListItem

Parameter	Type	Description
topic	String	The name of the topic.
subString	String	The subscription string.

3.1.6.5.5. /json/group/consumerUpdate

Configures the read permissions on messages for a consumer group that is identified by a group ID.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the group ID you want to configure resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
groupId	String	Yes	The group ID for which you want to configure the read permissions on messages.
readEnable	Boolean	Yes	Specifies whether to grant the read permissions on messages.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.6. Consumption management

3.1.6.6.1. /json/consumer/accumulate

Queries the message accumulation of a consumer group. The queried information includes the number of unconsumed messages and the consumption latency of the consumer group.

Scenarios

You can call this operation to query the information about message accumulation in the production environment where you need to know the consumption progress of consumer groups. You can use the queried information to analyze message consumption and consumption latency.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the group ID you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
groupid	String	Yes	The group ID that you want to query.
detail	Boolean	No	Specifies whether to query the detailed information about message accumulation, including the message accumulation details of each queue.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The information about message accumulation.

Data

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
onLine	Boolean	Indicates whether the group was online. The group was online if one of the consumer instances that are identified by the group ID was online. Valid values: - true: online - false: offline
totalDiff	Long	The total number of accumulated messages in all the topics to which the group has subscribed.
consumeTps	Float	The total TPS for the messages that were received by the consumer instances that are identified by the group ID.
lastTimestamp	Long	The production time of the last message that was consumed by the consumer instances that are identified by the group ID.
delayTime	Long	The maximum message consumption latency of the consumer instances that are identified by the group ID.
DetailInTopicList	List(OnsConsumerAccumulateResponse.Data.DetailInTopicDo)	The detailed information about topics.

DetailInTopicList

Parameter	Type	Description
topic	String	The name of the topic.
totalDiff	Long	The number of accumulated messages in the topic.
lastTimestamp	Long	The production time of the last message that was consumed in the topic.
delayTime	Long	The maximum consumption latency of messages in the topic.

3.1.6.6.2. /json/consumer/status

Queries the detailed information about a group ID, including the subscription consistency, consumption TPS, load balancing, and consumer connection status.

Scenarios

Notice

- You can call this operation to troubleshoot abnormal consumption after you confirm whether messages are accumulated and whether consumers are online. You can check whether subscriptions are consistent for a group ID and whether load balancing is normal, and query the jstack information of online consumers.
- This operation uses a lot of backend methods for data aggregation. Therefore, the system responds slowly. We recommend that you do not frequently call this operation.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the group ID you want to query resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
groupid	String	Yes	The group ID that you want to query.
detail	Boolean	No	Specifies whether to query detailed information. Valid values: • true: queries detailed information. You can obtain the detailed information from the ConsumerConnectionInfoList and DetailInTopicList parameters in the response. • false: does not query detailed information. The ConsumerConnectionInfoList and DetailInTopicList parameters are empty in the response. Default value: false.

Parameter	Type	Required	Description
needJstack	Boolean	No	Specifies whether to display jstack information. Valid values: • true: displays jstack information. You can obtain the jstack information from the Jstack parameter in the response. • false: does not display jstack information. The Jstack parameter is empty in the response. Default value: false.  Note If you need to display jstack information, make sure that the detail parameter in the request is set to true.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	ConsumerStatusDo	The query results.

ConsumerStatusDo

Parameter	Type	Description
online	Boolean	Indicates whether the group was online.
totalDiff	Long	The total number of accumulated messages in the consumer group.
consumeTps	Long	The total consumption TPS.
lastTimestamp	Long	The latest consumption time.
delayTime	Long	The consumption latency.
consumeModel	Long	The consumption mode.
subscriptionSame	Boolean	Indicates whether subscriptions were consistent.
rebalanceOK	Boolean	Indicates whether load balancing was normal. Valid values: • true: normal • false: abnormal

Parameter	Type	Description
ConnectionSet	List(ConnectionDo)	The information about online consumers in the consumer group.
DetailInTopicList	List(DetailInTopicDo)	The consumption status of topics. If you need to query detailed information, make sure that the detail parameter in the request is set to true. Otherwise, the return value of this parameter is empty.
ConsumerConnectionInfoList	List(ConsumerConnectionInfoDo)	The detailed information about online consumers in the consumer group, including the jstack information and consumption response time (RT). If you need to query detailed information, make sure that the detail parameter in the request is set to true. Otherwise, the return value of this parameter is empty.

ConnectionDo

Parameter	Type	Description
clientId	String	The ID of the consumer instance.
clientAddr	String	The address and port of the consumer instance.
language	String	The language of the client where the consumer instance resides.
version	String	The version of the client where the consumer instance resides.
remoteIP	String	The host IP address or public IP address.

DetailInTopicDo

Parameter	Type	Description
topic	String	The name of the topic.
totalDiff	Long	The number of accumulated messages in the topic.
lastTimestamp	Long	The latest consumption time.

Parameter	Type	Description
delayTime	Long	The consumption latency.

ConsumerConnectionInfoDo

Parameter	Type	Description
clientId	String	The ID of the consumer instance.
connection	String	The connection information.
language	String	The language of the client where the consumer instance resides.
version	String	The version of the client where the consumer instance resides.
consumeModel	String	The consumption mode. Valid values: - CLUSTERING: clustering consumption - BROADCASTING: broadcasting consumption
consumeType	String	The way in which the consumer instance consumes messages. Valid values: - PUSH: The RocketMQ broker pushes messages to the consumer instance. - PULL: The consumer instance pulls messages from the RocketMQ broker.
threadCount	Integer	The number of consumption threads.
startTimeStamp	Long	The start time of consumption.
lastTimeStamp	Long	The latest consumption time.
SubscriptionSet	List(SubscriptionData)	The information about subscriptions.
RunningDataList	List(ConsumerRunningDataDo)	The real-time statistics.
Jstack	List(ThreadTrackDo)	The jstack information.

SubscriptionData

Parameter	Type	Description
topic	String	The name of the topic to which the group has subscribed.
subString	String	The subscription string of the topic.
subVersion	Long	The subscription version. The value is automatically incremented and of the LONG type.
tagsSet	List(String)	The information about the tags to which the group has subscribed.

ConsumerRunningDataDo

Parameter	Type	Description
groupId	String	The group ID of the consumer.
topic	String	The name of the topic to which the group has subscribed.
rt	Float	The consumption RT. Unit: milliseconds.
okTps	Float	The TPS of successful message consumption.
failedTps	Float	The TPS of message consumption failures.
failedCountPerHour	Long	The number of messages that failed to be consumed per hour.

ThreadTrackDo

Parameter	Type	Description
thread	String	The name of the thread.
trackList	List(String)	The jstack information.

3.1.6.6.3. /json/consumer/getConnection

Queries the connection information about the consumer instances that are identified by a group ID.

Scenarios

You can call this operation to check whether the consumer instances that are identified by a group ID are online and query the detailed connection information.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the group ID you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
groupid	String	Yes	The group ID that you want to query.

Response parameters

Parameter	Type	Description
data	List(ConnectionDo)	The connection information for the group ID.

ConnectionDo

Name	Type	Description
clientId	String	The ID of the consumer instance.
clientAddr	String	The address and port of the consumer instance.
language	String	The language of the client where the consumer instance resides.
version	String	The version of the client where the consumer instance resides.
remoteIP	String	The host IP address or public IP address.

3.1.6.6.4. /json/consumer/timespan

Queries the timestamps of messages and the latest time of consumption in a topic to which a group has subscribed.

Scenarios

You can call this operation to query the latest time and earliest time of messages in a topic on a RocketMQ broker, and the last consumption time of consumers. This operation is used to display the consumption progress, and can be used together with the `/json/consumer/accumulate` operation. For more information, see [Query consumption accumulation \(/json/consumer/accumulate\)](#).

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance where the group ID you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
groupid	String	Yes	The group ID that you want to query.
topic	String	Yes	The topic to which the group has subscribed.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The query results.

Data

Parameter	Type	Description
topic	String	The name of the topic that you queried.
minTimeStamp	Long	The time when the latest message was stored in the topic.
maxTimeStamp	Long	The time when the earliest message was stored in the topic.
consumeTimeStamp	Long	The latest time when the group consumed a message in the topic.

3.1.6.6.5. /json/consumer/resetoffset

Resets a consumer offset to a timestamp for a group ID.

Scenarios

You can call this operation to clear accumulated messages or reset the consumption progress. You can clear accumulated messages by using one of the following methods:

- Clear all messages.
- Clear the messages that are sent before a specified point in time.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the group ID you want to query resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
groupId	String	Yes	The group ID that you want to query.

Parameter	Type	Required	Description
topic	String	Yes	The name of the topic for which you want to reset the consumer offset.
type	Integer	Yes	The method that is used to clear accumulated messages. Valid values: • 0: clears all messages. • 1: clears the messages that are sent before a specified point in time.
resetTime	Long	No	The timestamp to which the consumption progress is reset. This parameter takes effect only when the type parameter is set to 1.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

3.1.6.7. Message query

3.1.6.7.1. /json/message/trace

Queries whether a message has been consumed based on the message ID.

Scenarios

This operation is implemented based on the internal offset mechanism of Message Queue for Apache RocketMQ. In most cases, the results are valid. However, if you have reset the consumer offset or cleared messages, the results may be invalid.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the message you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
topic	String	Yes	The topic to which the message belongs.
msgId	String	Yes	The ID of the message.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List(MessageTrack)	The query results.

MessageTrack

Parameter	Type	Description
instanceId	String	The ID of the instance.
consumerGroup	String	The name of the consumer group.

Parameter	Type	Description
trackType	String	The status of the message. Valid values: • CONSUMED: The message was consumed. • CONSUMED_BUT_FILTERED: The message was filtered. • NOT_CONSUME_YET: The message was not consumed. • NOT_ONLINE: The consumer instance was offline. • UNKNOWN: Other unknown issues occurred.
exceptionDesc	String	The description of an exception.

3.1.6.7.2. /json/message/getById

Queries a message based on the message ID and queries whether the message has been consumed. The queried information about the message includes the sending time, RocketMQ broker, and properties such as the key and tag.

Scenarios

You can call this operation to query the information about a message based on the message ID. This is an exact match. You can obtain the message ID from the `SendResult` parameter that is returned after the message is sent. Therefore, you must store the results that are returned after each message is sent.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the message you want to query resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .

Parameter	Type	Required	Description
msgId	String	Yes	The ID of the message that you want to query.
topic	String	Yes	The topic to which the message belongs.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The query results.

Data

Parameter	Type	Description
topic	String	The topic to which the message belongs.
propertyList	List(MessageProperty)	The properties of the message.
body	String	The body of the message.
storeSize	Integer	The size of the message.
bornTimestamp	Long	The time when the message was generated.
bornHost	String	The producer that generated the message.
storeTimestamp	Long	The time when the RocketMQ broker stored the message.
storeHost	String	The RocketMQ broker that stored the message.
msgId	String	The ID of the message.
bodyCRC	Integer	The cyclic redundancy check (CRC) value of the message body.
reconsumeTimes	Integer	The number of message consumption retries.

3.1.6.7.3. /json/message/getByKey

Queries messages by using fuzzy match based on the topic and message key.

Scenarios

You can call this operation to query the information about messages based on the message key. This is a fuzzy match. A key may not be unique for a message. Therefore, the return results may contain multiple messages.

This operation is applicable to scenarios where you cannot obtain message IDs. In these scenarios, perform the following steps:

1. Call this operation to query message IDs by using fuzzy match.
2. Call the `/json/message/getById` operation to query a message based on the message ID. For more information, see [Query messages by message ID \(/json/message/getById\)](#).

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the messages you want to query reside. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The topic to which the messages belong.
key	String	Yes	The key of the messages.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	List(OnsRestMessageDo)	The query results.

OnsRestMessageDo

Parameter	Type	Description
topic	String	The topic to which the message belongs.
propertyList	List(MessageProperty)	The properties of the message.
body	String	The body of the message.
storeSize	Integer	The size of the message.
bornTimestamp	Long	The time when the message was generated.
bornHost	String	The producer that generated the message.
storeTimestamp	Long	The time when the RocketMQ broker stored the message.
storeHost	String	The RocketMQ broker that stored the message.
msgId	String	The ID of the message.
bodyCRC	Integer	The CRC value of the message body.
reconsumeTimes	Integer	The number of message consumption retries.

3.1.6.7.4. /json/message/getByPagedTopic

Queries all messages in a topic within a period of time by page.

Scenarios

You can call this operation to query messages based on the topic and time range. This is a range query. The query results are displayed on multiple pages. This operation is applicable only when message IDs and message keys are unavailable. However, we recommend that you shorten the time range to query. Otherwise, excessive messages may be returned and affect your troubleshooting efficiency.

To call this operation, perform the following steps:

1. Specify the topic, start time, end time, and size of each page to query messages by page. If matched messages are found, the first page of the messages, total number of pages, and query task ID are returned by default.
2. Query messages that are not displayed on the first page based on the returned query task ID for the first query. When you query the messages, specify the query task ID and current page number.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the messages you want to query reside. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The topic to which the messages belong.
beginTime	Long	Yes	The beginning of the time range to query.
endTime	Long	Yes	The end of the time range to query.
taskId	String	No	The ID of the query task. This parameter is not required for the first query but is required to query messages subsequently. You can obtain the value of this parameter from the previous return results.
currentPage	Integer	Yes	The number of the page to return. Valid values: 1 to 50.
pageSize	Integer	No	The number of entries to return on each page. Valid values: 5 to 50. Default value: 20.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
MsgFoundDo	MsgFoundDo	The query results.

MsgFoundDo

Parameter	Type	Description
taskId	String	The ID of the query task that is returned for the first query. This ID is used to query messages subsequently.
maxPageCount	Long	The total number of pages returned.
currentPage	Long	The page number of the returned page.
data	List(OnsRestMessageDo)	The information about the messages that are displayed on the current page. The return results are the same as those of the <code>/json/message/getById</code> operation.

3.1.6.8. Publishing and subscription statistics

3.1.6.8.1. /json/trend/topicInputTps

Queries the statistics about messages that are written to a topic within a period of time.

Scenarios

You can call this operation to query online message statistics for a specified topic, and query the number of messages that are written or the TPS curve within a specified period of time.

 **Notice** If the message volume is small and messages are unevenly distributed for your application, the TPS curve may be flat. We recommend that you query the total number of messages that are written to the topic.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the topic you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The name of the topic that you want to query.
beginTime	Long	Yes	The beginning of the time range to query. Unit: milliseconds.
endTime	Long	Yes	The end of the time range to query. Unit: milliseconds.
period	Long	Yes	The sampling period. Unit: minutes. Valid values: 1, 5, and 10.
type	Integer	Yes	The type of statistics that you want to query. Valid values: 0: queries the number of messages that are written to the specified topic within the specified period of time. 1: queries the TPS curve within the specified period of time.

Response parameters

Parameter	Type	Description
data	Data	The returned data.

Data

Parameter	Type	Description
title	String	The name of the table.
records	List(StatsDataDo)	The collected statistics.

StatsDataDo

Parameter	Type	Description
x	Long	The x-axis that represents the timestamp. Unit: milliseconds.
y	Float	The y-axis that represents the data, which is the TPS or total number of messages.

3.1.6.8.2. /json/trend/groupOutputTps

Queries the statistics about messages that are consumed by a group within a period of time.

Scenarios

When the system generates consumption statistics online, you can call this operation to query the following statistics based on a specified group ID:

- The statistical curve of the total number of consumed messages
- The statistical curve of the TPS for message consumption

 **Notice** If the message volume is small and messages are unevenly distributed for your application, the TPS curve may be flat. We recommend that you query the total number of consumed messages.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance where the group ID you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
topic	String	Yes	The name of the topic that you want to query.
groupId	String	Yes	The group ID that you want to query.
beginTime	Long	Yes	The beginning of the time range to query. Unit: milliseconds.
endTime	Long	Yes	The end of the time range to query. Unit: milliseconds.
period	Long	Yes	The sampling period. Unit: minutes. Valid values: 1, 5, and 10.
type	Integer	Yes	The type of statistics that you want to query. Valid values: 0: queries the statistical curve of the total number of consumed messages. 1: queries the statistical curve of the TPS for message consumption.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

Parameter	Type	Description
data	Data	The returned data.

Data

Parameter	Type	Description
title	String	The name of the table.
records	List(StatsDataDo)	The collected statistics.

StatsDataDo

Parameter	Type	Description
x	Long	The x-axis that represents the timestamp. Unit: milliseconds.
y	Float	The y-axis that represents the data, which is the TPS or total number of messages.

3.1.6.9. Authorization management

3.1.6.9.1. /json/empower/create

Authorizes another Apsara Stack tenant account to use a topic, including the permissions to publish and subscribe to messages.

Scenarios

In Message Queue for Apache RocketMQ, a topic belongs to the owner account that is used to create the topic. If another Apsara Stack tenant account needs to use the topic, the owner can call this operation to grant the permissions to publish and subscribe to messages to the Apsara Stack tenant account.

Note

- Three types of permissions are supported for authorization: publishing, subscription, and publishing and subscription.
- The account that is used to grant permissions must be the owner account of the topic. Permissions are granted to an Apsara Stack tenant account.
- After permissions are granted, the Apsara Stack tenant account still needs to create its own group ID for subscribing to and receiving messages instead of using the group ID that is created by the owner account.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.
empowerUser	Long	Yes	The user ID of the Apsara Stack tenant account to be authorized.
topic	String	Yes	The topic whose permissions you want to grant. The topic must belong to you.
relation	Integer	Yes	The type of permission to be granted. Valid values: 2: the permissions to publish messages 4: the permissions to subscribe to messages 6: the permissions to publish and subscribe to messages

Response parameters

The returned data is null.

3.1.6.9.2. /json/empower/createBySub

Authorizes a RAM user to use a topic, including the permissions to publish and subscribe to messages.

Scenarios

In Message Queue for Apache RocketMQ, a topic belongs to the owner account that is used to create the topic. If a RAM user needs to use the topic, the owner can call this operation to grant the permissions to publish and subscribe to messages to the RAM user.

Note

- Three types of permissions are supported for authorization: publishing, subscription, and publishing and subscription.
- The account that is used to grant permissions must be the owner account of the topic. Permissions are granted to a RAM user.
- After permissions are granted, the RAM user still needs to create its own group ID for subscribing to and receiving messages instead of using the group ID that is created by the owner account.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.
empowerUser	Long	Yes	The ID of the RAM user to be authorized.

Parameter	Type	Required	Description
topic	String	Yes	The topic whose permissions you want to grant. The topic must belong to you.
relation	Integer	Yes	The type of permission to be granted. Valid values: 2: the permissions to publish messages 4: the permissions to subscribe to messages 6: the permissions to publish and subscribe to messages

Response parameters

The returned data is null.

3.1.6.9.3. /json/empower/delete

Deletes an existing authorization to revoke the topic access permissions that are granted to an account.

Scenarios

After the permissions to access a topic are granted to an account, the owner of the topic can call this operation to revoke the granted permissions.

Note

- The account that is used to call this operation must be the owner account of the topic.
- The authorization information to be deleted must exist.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceId	String	Yes	The ID of the instance.
empowerUser	Long	Yes	The user ID of the authorized account. A RAM user ID is supported.
topic	String	Yes	The topic whose permissions are granted. The topic must belong to you.

Response parameters

The returned data is null.

3.1.6.9.4. /json/empower/list

Queries the authorization information about a topic.

Scenarios

The owner of a topic can query whether the topic access permissions are granted to a specific account and query the type of granted permission.

Request parameters

For more information about common request parameters, see [Common request parameters](#).

Parameter	Type	Required	Description
instanceld	String	Yes	The ID of the instance.
empowerUser	Long	Yes	The user ID of the authorized account. A RAM user ID is supported.
topic	String	Yes	The topic whose permissions are granted. The topic must belong to you.

Response parameters

The authorization information about the topic is returned in the JSON format, including the authorized account and type of granted permission.

3.1.6.10. Dead-letter queue management

3.1.6.10.1. /json/message/getByIdInDLQ

Queries a dead-letter message based on the message ID. The queried information about the dead-letter message includes the storage time, message body, and properties such as the key and tag.

Description

You can call this operation to query the information about a dead-letter message based on the message ID. This is an exact match. You can obtain the message ID from the `SendResult` parameter that is returned after the message is sent. You can also call the `/json/message/getByPagedDLQ` operation to query message IDs.

Request parameters

Parameter	Type	Required	Description
-----------	------	----------	-------------

Parameter	Type	Required	Description
InstanceId	String	Yes	The ID of the instance where the message you want to query resides. - If your instance has no namespace, you can leave this parameter unspecified. - If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
GroupId	String	Yes	The group ID that you want to query.
MsgId	String	Yes	The ID of the message that you want to query.

Response parameters

Data

Parameter	Type	Description
Topic	String	The topic to which the message belongs.
PropertyList	List(MessageProperty)	The properties of the message.
Body	String	The body of the message.
StoreSize	Integer	The size of the message.
BornTimestamp	Long	The time when the message was generated.
BornHost	String	The producer that generated the message.
StoreTimestamp	Long	The time when the RocketMQ broker stored the message.
StoreHost	String	The RocketMQ broker that stored the message.
MsgId	String	The ID of the message.

Parameter	Type	Description
BodyCRC	Integer	The CRC value of the message body.
ReconsumeTimes	Integer	The number of message consumption retries.

3.1.6.10.2. /json/message/getByPagedDLQ

Queries all dead-letter messages in a group within a period of time by page.

Description

You can call this operation to query dead-letter messages based on the group ID. This is a range query. The query results are displayed on multiple pages. This operation is applicable only when message IDs are unavailable. However, we recommend that you shorten the time range to query. Otherwise, excessive messages may be returned and affect your troubleshooting efficiency.

To call this operation, perform the following steps:

1. Specify the group ID, start time, end time, and size of each page to query messages by page. If matched messages are found, the first page of the messages, total number of pages, and query task ID are returned by default.
2. Query messages that are not displayed on the first page based on the returned query task ID for the first query. When you query the messages, specify the query task ID and current page number.

Request parameters

Parameter	Type	Required	Description
Instanceld	String	Yes	The ID of the instance where the messages you want to query reside. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates.
GroupId	String	Yes	The group ID that you want to query.
BeginTime	Long	Yes	The beginning of the time range to query.

Parameter	Type	Required	Description
EndTime	Long	Yes	The end of the time range to query.
TaskId	String	No	The ID of the query task. This parameter is not required for the first query but is required to query messages subsequently. You can obtain the value of this parameter from the previous return results.
CurrentPage	Integer	Yes	The number of the page to return. Valid values: 1 to 50.
PageSize	Integer	No	The number of entries to return on each page. Valid values: 5 to 50. Default value: 20.

Response parameters

Parameter	Type	Description
MsgFoundDo	MsgFoundDo	The query results.

MsgFoundDo

Parameter	Type	Description
TaskId	String	The ID of the query task that is returned for the first query. This ID is used to query messages subsequently.
MaxPageCount	Long	The total number of pages returned.
CurrentPage	Long	The page number of the returned page.
Data	List(OnsRestMessageDo)	The information about the messages that are displayed on the current page. The return results are the same as those of the <code>/json/message/getByIdInDLQ</code> operation.

3.1.6.10.3. /json/message/resendByIdInDLQ

Resends a dead-letter message based on the message ID so that the message can be consumed again by the same consumer.

Description

You can call this operation to allow a dead-letter message to be consumed by the same consumer again.

A dead-letter message is a message that still fails to be consumed after the number of consumption retries reaches the upper limit. If the message still cannot be consumed after you resend it, a message with the same message ID is added to the corresponding dead-letter queue. You can query the message ID on the Dead-letter Queues page in the Message Queue for Apache RocketMQ console or by calling API operations. You can determine the number of consumption failures for a message after you resend it, based on the number of messages with the same message ID.

Notice

- A dead-letter message is a message that still fails to be consumed after the number of consumption retries reaches the upper limit. This generally means that the consumption logic may have errors. Therefore, first identify the causes of the consumption failures, and then call this operation to resend the message.
- Message Queue for Apache RocketMQ does not manage the status of dead-letter messages after you resend them. Messages that are consumed after you call this operation are not removed from corresponding dead-letter queues. You must manually manage dead-letter messages and determine whether to resend a dead-letter message. This way, you do not need to repeatedly resend dead-letter messages that are consumed.

Request parameters

Parameter	Type	Required	Description
Instanceld	String	Yes	The ID of the instance where the message you want to query resides. • If your instance has no namespace, you can leave this parameter unspecified. • If your instance has a namespace, you must specify this parameter. For more information about namespaces, see Updates .
MsgId	String	Yes	The ID of the message that you want to query.

Parameter	Type	Required	Description
GroupId	String	Yes	The group ID that you want to query.

Response parameters

For more information about common response parameters, see [Common response parameters](#).

4.SDK reference

4.1. 文档标题缺失，请补全后重新导出

4.1.1. Overview

Message Queue for Apache RocketMQ is an important component of distributed systems. This service supports asynchronous communication for application decoupling and load shifting, and provides high performance, high availability, high scalability, and eventual consistency.

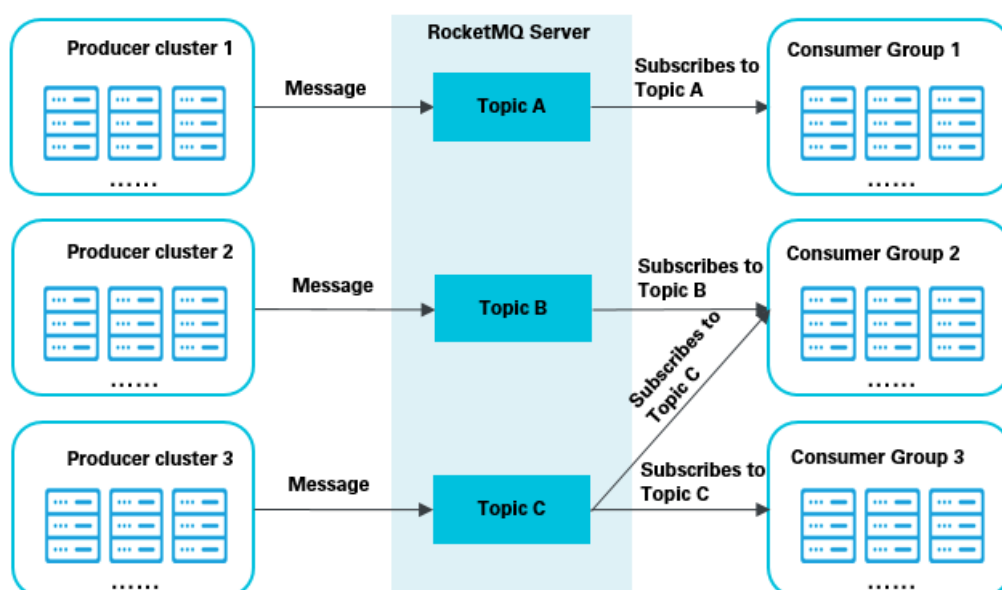
This document describes how to manage a message queuing service between RocketMQ applications and the message queuing server during data transmission. This document provides the specifications that the service complies with and the common API operations that are used to manage the service.

This document is suitable for scenarios in which messages are transmitted over the TCP protocol by using Message Queue for Apache RocketMQ SDK for Java.

4.1.2. Messaging model

Message Queue for Apache RocketMQ supports the publish-subscribe model. A producer can publish a message to a broker, and multiple consumers can subscribe to a topic on the broker. The following figure shows the messaging model.

Publish-subscribe model



4.1.3. Updates

This topic describes the updates of Message Queue for Apache RocketMQ from V3.8.0 to V3.8.1 to help you get started with the updated version.

Optimization of resource isolation by instance

Message Queue for Apache RocketMQ provides instances for multi-tenancy isolation. Each user can purchase multiple instances and logically isolate them from each other.

To ensure the compatibility with the existing resources of existing users, Message Queue for Apache RocketMQ provides the following types of instances and namespaces:

- Default instances, which are compatible with the existing resources of existing users
 - This type of instance has no separate namespace. Resource names must be globally unique within and across all instances.
 - By default, an instance without a namespace is automatically generated for the existing resources of each existing user. If no existing resources are available, you can create at most one instance without a namespace.
 - You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxxx");
// Compatible configuration, which is not recommended. We recommend that you update this configuration to the recommended configuration:
properties.put(PropertyKeyConst.ONSAddr, "xxxx");
```

- New instances
 - A new instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances.
 - You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
```

- A RocketMQ client must be updated to the following latest versions for different programming languages:
 - Java: **V1.8.7.1.Final**
 - C and C++: **V2.0.0**
 - .NET: **V1.1.3**

Optimization of resource application

Previously, Message Queue for Apache RocketMQ resources consisted of topics, producer IDs, and consumer IDs. Each two of the resources have a many-to-many relationship, which was difficult to comprehend. Each time you created a topic, you must associate the topic with a producer ID and a consumer ID. This process was too complex for medium- and large-sized enterprise customers.

To optimize user experience and help new users get started, the resource application process has been simplified.

The resource application process has been optimized in the following aspects:

- Topic management, which is unchanged
 - You need to apply for a topic. A topic is used to classify messages. It is the primary classifier.
- Group management

- You do not need to apply for a producer ID. Producer IDs and consumer IDs are integrated into group IDs. In the Message Queue for Apache RocketMQ console, the Producers module has been removed. The Producers and Consumers modules have been integrated into the Groups module.
- You do not need to associate a producer ID or consumer ID with a topic. Instead, you need only to apply for a group ID and associate it with a topic in the code.
- Compatibility:
 - The list of producer IDs is no longer displayed. This does not affect the current services.
 - The consumer IDs that start with CID- or CID_ and that you have applied for can still be used and can be set in the `PropertyKeyConst.ConsumerId` or `PropertyKeyConst.GROUP_ID` parameter of the code.
- Sample code

Note

- We recommend that you update a RocketMQ client to the following latest versions for different programming languages:
 - Java: **V1.8.7.1.Final**
 - C and C++: **V2.0.0**
 - .NET: **V1.1.3**
- Existing producer IDs or consumer IDs can still be used and do not affect the current services. However, we recommend that you update your instance configuration to the recommended configuration.

- Recommended configuration: Integrate producer IDs and consumer IDs into group IDs.

```
// Set the PropertyKeyConst.GROUP_ID parameter. The original PropertyKeyConst.ProducerId and PropertyKeyConst.ConsumerId parameters are deprecated.
properties.put(PropertyKeyConst.GROUP_ID, "The original CID-XXX or the GID-XXX");
```

- Compatible configuration: Use a producer ID to identify a producer and a consumer ID to identify a consumer.

```
// When you create a producer, you must set the PropertyKeyConst.ProducerId parameter.
properties.put(PropertyKeyConst.ProducerId, "The original PID-XXX or the GID-XXX");
// When you create a consumer, you must set the PropertyKeyConst.ConsumerId parameter.
properties.put(PropertyKeyConst.ConsumerId, "The original CID-XXX or the GID-XXX");
```

4.1.4. Transport protocol and character encoding

The underlying Message Queue for RocketMQ (RocketMQ) SDK provides services through the TCP protocol, and user requests and responses use UTF-8 character encoding.

Transport layer encryption

RocketMQ provides support for Transport Layer Security (TLS). You can use the SDK configuration to determine whether to use encrypted transport layer links on the premise that you obtain support from the RocketMQ broker.

Configuration method: When starting the client program, set the JVM attributes to determine whether to use the encrypted transport layer link. The startup command is `java -Dtls.enable=true`. By default, TLS links are disabled. If you set the value to true, encrypted transport layer links are enabled.

4.1.5. Common classes and methods

4.1.5.1. PropertyKeyConst

The following table lists the main fields of this class.

Parameter	Type	Default value	Description
AccessKey	String	-	The AccessKey ID that you created in the Apsara Uni-manager Management Console for identity verification.
SecretKey	String	-	The AccessKey secret that you created in the Apsara Uni-manager Management Console for identity verification.
OnsChannel	String	ALIYUN	The user channel. Valid values: ALIYUN and CLOUD. ALIYUN stands for Apsara Stack. CLOUD stands for CloudTmall.
NAMESRV_ADDR	String	-	The TCP endpoint that is used to connect to the Message Queue for Apache RocketMQ instance.
GROUP_ID	String	-	The group ID that you created in the Message Queue for Apache RocketMQ console. This group ID identifies a type of producers or consumers.
INSTANCE_ID	String	-	The ID of the Message Queue for Apache RocketMQ instance. - If your instance is not assigned a namespace, you do not need to configure this parameter. - If your instance is assigned a namespace, you must configure this parameter. For more information about namespaces, see Updates. Make sure that you use the TCP client SDK for Java 1.8.4.Final or later.
MessageModel	String	CLUSTERING	The consumption mode for the consumer instance. Valid values: CLUSTERING and BROADCASTING.

Parameter	Type	Default value	Description
ConsumeThreadNums	String	64	The number of consumer threads.
MaxReconsumeTimes	String	16	The maximum number of delivery retries that can be performed when a message fails to be consumed.
ConsumeTimeout	String	15	The timeout period for consuming each message. If the time for consuming a message exceeds the specified timeout period, the message fails to be consumed and is redelivered after a retry interval. Configure an appropriate value for each type of application. Unit: minute.
ConsumeMessageBatchMaxSize	String	1	The maximum number of messages that can be consumed in each batch. The actual number of consumed messages in a batch can be smaller than the value of this parameter. The value must be an integer from 1 to 32. The default value is 1.
CheckImmunityTimeInSeconds	String	30	The shortest time for which the system waits before checking the status of a transactional message for the first time. Unit: seconds.
shardingKey	String	-	The partition key that is used to determine the partitions to which ordered messages are distributed.
suspendTimeMillis	String	3000	The retry interval for ordered messages that fail to be consumed. Unit: milliseconds.

4.1.5.2. ONSFactory

4.1.5.2.1. createConsumer

- Syntax: static Consumer createConsumer(Properties properties)
- Description: You can call this operation to create a consumer instance that consumes multiple types of messages in Message Queue for Apache RocketMQ. These message types include normal messages, scheduled messages, delayed messages, and transactional messages.
- Request parameter

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
properties	Properties	For more information, see the description of <code>PropertyKeyConst</code>. Required properties: <code>PropertyKeyConst.NAMESRV_ADDR</code>, <code>PropertyKeyConst.ConsumerId</code>, <code>PropertyKeyConst.AccessKey</code>, <code>PropertyKeyConst.INSTANCE_ID</code>, and <code>PropertyKeyConst.SecretKey</code>.  Note ◦ If your instance is not assigned a namespace, you do not need to configure <code>PropertyKeyConst.INSTANCE_ID</code>. If you configure <code>PropertyKeyConst.INSTANCE_ID</code>, unexpected issues can occur. ◦ If your instance is assigned a namespace, you must configure <code>PropertyKeyConst.INSTANCE_ID</code>. For more information about namespaces, see Updates. Optional properties: <code>PropertyKeyConst.ConsumeThreadNums</code> and <code>PropertyKeyConst.ConsumeTimeout</code>.

- Response parameter

Parameter	Description
Consumer	The consumer instance that consumes multiple types of messages in Message Queue for Apache RocketMQ. These message types include normal messages, scheduled messages, transactional messages, and delayed messages.

4.1.5.2.2. createBatchConsumer

- Syntax: `static BatchConsumer createBatchConsumer(Properties properties)`
- Description: You can call this operation to create consumer instances in Message Queue for RocketMQ (RocketMQ) in batches.
- Request parameters:

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
properties	Properties	For more information, see the description of PropertyKeyConst. Required properties: PropertyKeyConst.NAMESRV_ADDR, PropertyKeyConst.GROUP_ID, PropertyKeyConst.AccessKey, and PropertyKeyConst.SecretKey Optional properties: PropertyKeyConst.ConsumeThreadNums, PropertyKeyConst.ConsumeTimeout, and PropertyKeyConst.ConsumeMessageBatchMaxSize

- Response:

Type	Description
BatchConsumer	BatchConsumer

4.1.5.2.3. createOrderedConsumer

- Syntax: static OrderConsumer createOrderedConsumer(Properties properties)
- Description: You can call this operation to create a consumer instance for consuming ordered messages in Message Queue for RocketMQ (RocketMQ).
- Request parameters:

Parameter	Type	Description
properties	properties	For more information, see the description of PropertyKeyConst. Required properties: PropertyKeyConst.NAMESRV_ADDR, PropertyKeyConst.GROUP_ID, PropertyKeyConst.AccessKey, and PropertyKeyConst.SecretKey Optional properties: PropertyKeyConst.ConsumeThreadNums and PropertyKeyConst.ConsumeTimeout

- Response:

Type	Description
OrderConsumer	The consumer instance of ordered messages in RocketMQ.

4.1.5.2.4. createOrderProducer

- Syntax: static OrderProducer createOrderProducer(Properties properties)
- Description: You can call this operation to create a producer instance for sending ordered messages in Message Queue for RocketMQ (RocketMQ).
- Request parameters:

Parameter	Type	Description
properties	Properties	For more information, see the description of <code>PropertyKeyConst</code>. Required properties: <code>PropertyKeyConst.NAMESRV_ADDR</code>, <code>PropertyKeyConst.GROUP_ID</code>, <code>PropertyKeyConst.AccessKey</code>, <code>PropertyKeyConst.SecretKey</code>, and <code>PropertyKeyConst.ShardingKey</code> Optional properties: <code>PropertyKeyConst.SendMsgTimeoutMillis</code>

- Response:

Type	Description
OrderProducer	The producer instance of ordered messages in RocketMQ.

4.1.5.2.5. createProducer

- Syntax: static Producer createProducer(Properties properties)
- Description: You can call this operation to create a producer instance that publishes normal messages, scheduled messages, and delayed messages in Message Queue for Apache RocketMQ.
- Request parameter

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
properties	Properties	For more information, see the description of <code>PropertyKeyConst</code>. Required properties: <code>PropertyKeyConst.NAMESRV_ADDR</code>, <code>PropertyKeyConst.GROUP_ID</code>, <code>PropertyKeyConst.AccessKey</code>, <code>PropertyKeyConst.INSTANCE_ID</code>, and <code>PropertyKeyConst.SecretKey</code>.  Note ◦ If your instance is not assigned a namespace, you do not need to configure <code>PropertyKeyConst.INSTANCE_ID</code>. If you configure <code>PropertyKeyConst.INSTANCE_ID</code>, unexpected issues can occur. ◦ If your instance is assigned a namespace, you must configure <code>PropertyKeyConst.INSTANCE_ID</code>. For more information about namespaces, see Updates. Optional properties: <code>PropertyKeyConst.SendMsgTimeoutMillis</code>.

- Response parameter

Parameter	Description
Producer	The producer instance that publishes normal messages, scheduled messages, and delayed messages in Message Queue for Apache RocketMQ.

4.1.5.2.6. createTransactionProducer

- Syntax: `static TransactionProducer createTransactionProducer(Properties properties, LocalTransactionChecker checker)`
- Description: You can call this operation to create a producer instance for sending distributed transactional messages in Message Queue for RocketMQ (RocketMQ).
- Request parameters:

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
properties	Properties	For more information, see the description of <code>PropertyKeyConst</code>. Required properties: <code>PropertyKeyConst.NAMESRV_ADDR</code>, <code>PropertyKeyConst.GROUP_ID</code>, <code>PropertyKeyConst.AccessKey</code>, and <code>PropertyKeyConst.SecretKey</code> Optional properties: <code>PropertyKeyConst.SendMsgTimeoutMillis</code> and <code>PropertyKeyConst.CheckImmunityTimeInSeconds</code>
checker	<code>LocalTransactionChecker</code>	For more information, see the description of <code>LocalTransactionChecker</code> .

- Response:

Type	Description
<code>TransactionProducer</code>	The producer instance of transactional messages in RocketMQ.

4.1.5.3. Message

4.1.5.3.1. Overview

It is a message that you want to send. A valid message must contain the topic, body, optional tags, and message key.

Note: The length of the custom key-value pair for a message is not constrained, but all custom key-value pairs cannot exceed 32,767 bytes after they are serialized.

4.1.5.3.2. Default message construction

- Syntax: `Message()`
- Description: You can call this operation to create a message by using the default constructor.
- Request parameters: none

4.1.5.3.3. Message construction by using parameters (1)

- Syntax: `public Message(String topic, String tag, String key, byte body)`
- Description: You can call this operation to create a message by specifying parameters.
- Request parameters:

Parameter	Type	Description
topic	String	The topic of the message. It cannot exceed 255 characters in length. The message topic can contain uppercase letters (A-Z), lowercase letters (a-z), digits (0-9), hyphens (-), and underscores (_).

Parameter	Type	Description
tag	String	The message tag, which must be a valid identifier. The tag must be as short as possible and easy to understand.
key	String	The key of the message.
body	byte	The message body. It cannot exceed 4 MB by default, including message attributes that cannot exceed 8 KB. For more information about the message body, see the cluster deployment documentation.

4.1.5.3.4. Message construction by using parameters (2)

- Syntax: `public Message(String topic, String tags, byte body)`
- Description: You can call this operation to create a message by specifying parameters.
- Request parameters:

Parameter	Type	Description
topic	String	The topic of the message. It cannot exceed 255 characters in length. The message topic can contain uppercase letters (A-Z), lowercase letters (a-z), digits (0-9), hyphens (-), and underscores (_).
tags	String	The message tag, which must be a valid identifier. The tag must be as short as possible and easy to understand.
body	byte	The message body. It cannot exceed 4 MB by default, including message attributes that cannot exceed 8 KB. For more information about the message body, see the cluster deployment documentation.

4.1.5.3.5. putUserProperties

- Syntax: `public void putUserProperties(String key, String value)`
- Description: You can call this operation to add a custom property key-value pair that can be obtained when the message is consumed.
- Request parameters:

Parameter	Type	Description
key	String	The custom property key.
value	String	The custom property value.

- Response: none

4.1.5.3.6. getUserProperties

- Syntax: `public String getUserProperties(String key)`
- Description: You can call this operation to obtain the custom property value.
- Request parameters:

Parameter	Type	Description
key	String	The custom property key.

- Response:

Type	Description
String	The custom property value.

4.1.5.3.7. getTopic

- Syntax: `public String getTopic()`
- Description: You can call this operation to obtain the topic of the messages.
- Request parameters: none
- Response:

Type	Description
String	The topic of the messages.

4.1.5.3.8. setTopic

- Syntax: `public void setTopic(String topic)`
- Description: You can call this operation to set the topic of the messages.
- Request parameters:

Parameter	Type	Description
topic	String	The topic of the messages.

- Response: none

4.1.5.3.9. getTag

- Syntax: `public String getTag()`
- Description: You can call this operation to obtain the message tag.
- Request parameters: none
- Response:

Type	Description
String	The tag of the message.

4.1.5.3.10. setTag

- Syntax: `public void setTag(String tag)`
- Description: You can call this operation to set the message tag.
- Request parameters:

Parameter	Type	Description
tag	String	The tag of the message.

- Response: none

4.1.5.3.11. getKey

- Description: You can call this operation to obtain the message key.
- Syntax: `public String getKey()`
- Request parameters: none
- Response:

Type	Description
String	The key of the message.

4.1.5.3.12. setKey

- Syntax: `public void setKey(String key)`
- Description: You can call this operation to set the message key.
- Request parameters:

Parameter	Type	Description
key	String	The key of the message.

- Response: none

4.1.5.3.13. getMsgID

- Syntax: `public String getMsgID()`
- Description: You can call this operation to obtain the message ID.
- Request parameters: none
- Response:

Type	Description
String	The ID of the message.

4.1.5.3.14. setMsgID

- Syntax: `public void setMsgID(String msgid)`
- Description: You can call this operation to set the message ID.
- Request parameters:

Parameter	Type	Description
msgid	String	The ID of the message.

- Response: none

4.1.5.3.15. getReconsumeTimes

- Syntax: `public int getReconsumeTimes()`
- Description: You can call this operation to obtain the number of retries after a message has failed to be consumed.
- Request parameters: none
- Response:

Type	Description
String	The number of retries.

4.1.5.3.16. setReconsumeTimes

- Syntax: `public void setReconsumeTimes(int value)`
- Description: You can call this operation to set the number of retries to consume a message.
- Request parameters:

Parameter	Type	Description
value	int	The number of retries.

- Response: none

4.1.5.3.17. getBornTimestamp

- Syntax: `public long getBornTimestamp()`
- Description: You can call this operation to obtain the message production time.
- Request parameters: none
- Response:

Type	Description
String	The production time of the message.

4.1.5.3.18. setBornTimestamp

- Syntax: `public void setBornTimestamp(long value)`
- Description: You can call this operation to set the message production time.
- Request parameters:

Parameter	Type	Description
value	long	The production time of the message.

- Response: none

4.1.5.3.19. getBornHost

- Syntax: `public String getBornHost()`
- Description: You can call this operation to retrieve the message production host.
- Request parameters: none
- Response:

Type	Description
String	The host that produced the message.

4.1.5.3.20. setBornHost

- Syntax: `public void setBornHost(String value)`
- Description: You can call this operation to set the message production host.
- Request parameters:

Parameter	Type	Description
value	long	The host that produces the message.

- Response: none

4.1.5.3.21. getStartDeliverTime

- Syntax: `public long getStartDeliverTime()`
- Description: You can call this operation to obtain the start time for delivering scheduled messages.
- Request parameters: none
- Response:

Type	Description
String	The start time for delivering scheduled messages.

4.1.5.3.22. setStartDeliverTime

- Syntax: `public void setStartDeliverTime(long value)`
- Description: You can call this operation to set the delivery time (absolute time) of scheduled messages with a maximum latency of 40 days.
- Request parameters:

Parameter	Type	Description
value	long	The start time for delivering scheduled messages.

- Response: none

4.1.5.3.23. getShardingKey

- Syntax: `public long getShardingKey()`
- Description: You can call this operation to obtain the sharding key for ordered messages.
- Request parameters: none
- Response:

Type	Description
String	The sharding key for ordered messages.

4.1.5.3.24. setShardingKey

- Syntax: `public void setShardingKey(String value)`
- Description: You can call this operation to set the sharding key for ordered messages.
- Request parameters:

Parameter	Type	Description
String	long	The sharding key for ordered messages.

- Response: none

4.1.5.4. ONSClientException

4.1.5.4.1. Default ONSClientException construction

- Syntax: `public ONSClientException()`
- Description: Uses the default exception constructor to construct ONSClientException.

- Request parameters: none

4.1.5.4.2. ONSCliException construction by using parameters (1)

- Syntax: `public ONSCliException(String message)`
- Description: Uses the exception constructor to construct `ONSCliException` by using parameters.
- Request parameters

Parameter	Type	Description
message	String	The exception information that needs to be exposed.

4.1.5.4.3. ONSCliException construction by using parameters (2)

- Syntax: `public ONSCliException(Throwable cause)`
- Description: Uses the exception constructor to construct `ONSCliException` by using parameters.
- Request parameters

Parameter	Type	Description
message	String	The exception information that needs to be exposed.

4.1.5.4.4. ONSCliException construction by using parameters (3)

- Syntax: `public ONSCliException(String message, Throwable cause)`
- Description: Uses the exception constructor to construct `ONSCliException` by using parameters.
- Request parameters

Parameter	Type	Description
message	String	The exception information that needs to be exposed.
cause	Throwable	The exception information that needs to be exposed.

4.1.5.5. Message sending

4.1.5.5.1. Overview

This topic describes the methods for sending various types of messages in Message Queue for Apache RocketMQ. The following figure shows the methods and parameters for different message types.

Methods and parameters for publishing messages

□

4.1.5.5.2. SendResult

4.1.5.5.2.1. Default SendResult construction

- Syntax: `public SendResult()`
- Description: Uses the default constructor to create a message sending result object.
- Request parameters: none

4.1.5.5.2.2. getMessageId

- Syntax: `public long getMessageId()`
- Description: Queries the ID of a message that is sent.
- Request parameters: none
- Response parameters

Type	Description
String	The ID of the sent message.

4.1.5.5.2.3. getTopic

- Syntax: `public String getTopic()`
- Description: Queries the topic to which a message that is sent belongs.
- Request parameters: none
- Response parameters

Type	Description
String	The topic of the sent message.

4.1.5.5.2.4. toString

- Syntax: `public String toString()`
- Description: Serializes a message sending result.
- Request parameters: none
- Response parameters

Type	Description
String	The serialized string of the message sending result.

4.1.5.5.3. Producer

4.1.5.5.3.1. Overview

You can use a producer instance to send regular messages, scheduled messages, or delayed messages.

- Thread security: All implementation classes of a producer instance are thread-secure. Multiple threads can be used to send messages concurrently.
- A producer instance can send messages under multiple topics.
- In normal cases, a producer instance is efficient enough to complete all the message sending tasks by using relevant methods. No more instances need to be created.

Regular messages can be sent in reliable synchronous, reliable asynchronous, or one-way mode.

4.1.5.5.3.2. send

- Syntax: `public SendResult send(Message message)`
- Description: Sends a regular message, scheduled message, or delayed message in synchronous mode. The message is sent if no exception is thrown.
- Request parameters

Parameter	Type	Description
message	Message	The message that you want to send.

- Response parameters

Parameter	Description
SendResult	The sending result that is returned, including the message ID and topic.

4.1.5.5.3.3. sendOneway

- Syntax: `public void sendOneway(Message message)`
- Description: Sends a regular message, scheduled message, or delayed message in one-way mode. In this mode, if the RocketMQ broker does not respond, you cannot know whether the message is sent to the broker.
- Request parameters

Parameter	Type	Description
message	Message	The message that you want to send.

- Response parameters: none

4.1.5.5.3.4. sendAsync

- Syntax: `public void sendAsync(Message message, SendCallback sendCallback)`
- Description: Sends a regular message, scheduled message, or delayed message in asynchronous callback mode.
- Request parameters

Parameter	Type	Description
message	Message	The message that you want to send.
sendCallback	SendCallback	The callback function to be called after the message is sent.

- Response parameters: none

4.1.5.5.3.5. isStarted

- Syntax: `public boolean isStarted()`
- Description: Checks whether an instance is enabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.5.3.6. isClosed

- Syntax: `public boolean isClosed()`
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.5.3.7. start

- Syntax: `public void start()`
- Description: Enables a producer instance to send messages.

- Request parameters: none
- Response parameters: none

4.1.5.5.3.8. shutdown

- Syntax: `public void shutdown()`
- Description: Disables a producer instance from sending messages.
- Request parameters: none
- Response parameters: none

4.1.5.5.4. TransactionProducer

4.1.5.5.4.1. send

- Syntax: `SendResult send(Message message, LocalTransactionExecutor executor, Object arg)`
- Description: Sends a transactional message. The process of sending a transactional message consists of the following three steps:
 - The producer instance first sends a half message to the RocketMQ broker.
 - The producer instance starts a local transaction by using an executor.
 - The RocketMQ broker commits or rolls back the half message that is sent in Step 1 based on the result of the local transaction in Step 2.
- Request parameters

Parameter	Type	Description
message	Message	The transactional message that you want to send.
executor	LocalTransactionExecutor	The local transaction executor. For more information, see the <code>LocalTransactionExecutor</code> class.
arg	Object	A custom parameter. This parameter can be used for the local transaction executor.

- Response parameters

Parameter	Description
SendResult	The sending result that is returned, including the message ID and topic.

4.1.5.5.4.2. isStarted

- Syntax: `public boolean isStarted()`
- Description: Checks whether an instance is enabled.

- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.5.4.3. isClosed

- Syntax: `public boolean isClosed()`
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.5.4.4. start

- Syntax: `public void start()`
- Description: Enables a producer instance to send transactional messages.
- Request parameters: none
- Response parameters: none

4.1.5.5.4.5. shutdown

- Syntax: `public void shutdown()`
- Description: Disables a producer instance from sending transactional messages.
- Request parameters: none
- Response parameters: none

4.1.5.5.4.6. LocalTransactionExecuter

execute

- Syntax: `TransactionStatus execute(Message msg, Object arg)`
- Description: Starts a local transaction. This method is called by applications.
- Request parameters

Parameter	Type	Description
message	Message	The transactional message that you want to send.

Parameter	Type	Description
arg	Object	The custom parameter that is specified by the send method.

- Response parameters

Parameter	Description
TransactionStatus	The result of the local transaction that is returned. The transaction may be in the Commit, Rollback, or unknown state.

4.1.5.5.4.7. TransactionStatus

The result of a local transaction that corresponds to a half message. The transaction may be in the Commit, Rollback, or unknown state.

Enumeration

Value	Description
CommitTransaction	The transaction is committed. Then, the RocketMQ broker delivers the half message to the consumer.
RollbackTransaction	The transaction is rolled back. Then, the RocketMQ broker deletes the half message and does not deliver it to the consumer.
Unknow	The transaction is in the unknown state. Generally, this value is used when you are not sure whether the transaction is successful or fails. The RocketMQ broker periodically checks the transactions that are in the unknown state.

4.1.5.5.5. OrderProducer

4.1.5.5.5.1. send

- Syntax: `SendResult send(Message message, String shardingKey)`
- Description: Sends an ordered message.
- Request parameters

Parameter	Type	Description
message	Message	The ordered message that you want to send.
shardingKey	String	The selectivity factor for the ordered message. A specific message queue is selected based on the value of the shardingKey parameter.

- Response parameters

Parameter	Description
SendResult	The sending result that is returned, including the message ID and topic.

4.1.5.5.5.2. isStarted

- Syntax: public boolean isStarted()
- Description: Checks whether an instance is enabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.5.5.3. isClosed

- Syntax: public boolean isClosed()
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.5.5.4. start

- Syntax: public void start()
- Description: Enables a producer instance to send ordered messages.
- Request parameters: none
- Response parameters: none

4.1.5.5.5.5. shutdown

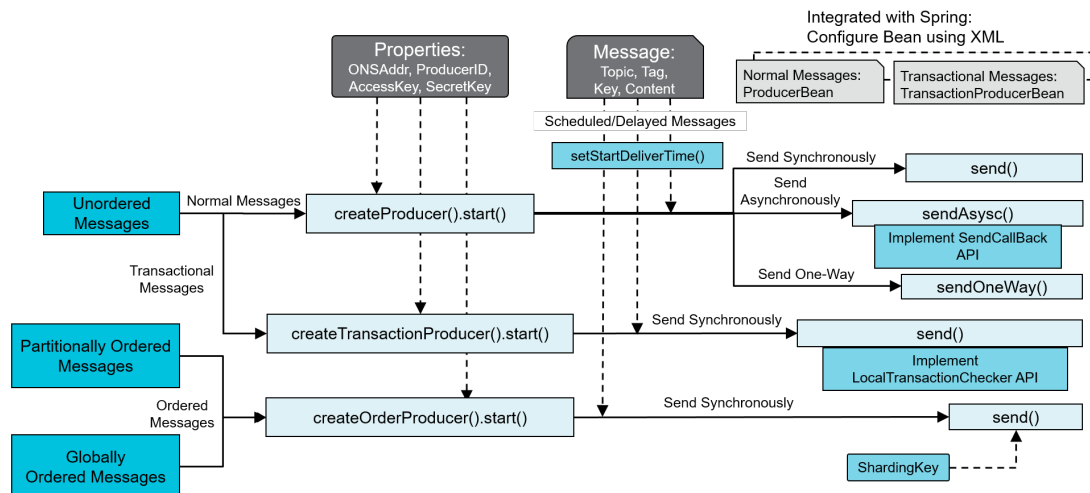
- Syntax: public void shutdown()
- Description: Disables a producer instance from sending ordered messages.
- Request parameters: none
- Response parameters: none

4.1.5.6. Message subscription

4.1.5.6.1. Overview

This topic describes the methods for subscribing to various types of messages in Message Queue for Apache RocketMQ. The following figure shows the methods and parameters for different message types.

Methods and parameters for subscribing to messages



4.1.5.6.2. Consumer

4.1.5.6.2.1. Overview

You can use a consumer instance to subscribe to regular messages, transactional messages, scheduled messages, or delayed messages.

4.1.5.6.2.2. subscribe

- Syntax: `public void subscribe(String topic, String subExpression, MessageListener listener)`
- Description: Subscribes to regular messages, transactional messages, scheduled messages, or delayed messages.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.
subExpression	String	The filter expression for subscribing to messages. The RocketMQ broker filters messages based on this expression. Only OR () operations are supported. Example 1: "tag1 tag2 tag3"; Example 2: When the subExpression parameter is set to null or *, all messages are subscribed to.
listener	MessageListener	The message listener for callbacks.

- Response parameters: none

4.1.5.6.2.3. unsubscribe

- Syntax: void unsubscribe(String topic)
- Description: Cancels the subscription to a specific topic.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.

- Response parameters: none

4.1.5.6.2.4. isStarted

- Syntax: public boolean isStarted()
- Description: Checks whether an instance is enabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.6.2.5. isClosed

- Syntax: public boolean isClosed()
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.6.2.6. start

- Syntax: public void start()
- Description: Enables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.2.7. shutdown

- Syntax: public void shutdown()
- Description: Disables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.3. MessageListener

4.1.5.6.3.1. Overview

A consumer registers a message listener to subscribe to messages.

Thread security requirements: A message listener can be concurrently used by multiple threads of a consumer. You must ensure the security of concurrency.

4.1.5.6.3.2. consume

- Syntax: Action consume(Message message, ConsumeContext context)
- Description: Consumes messages. This method is called by applications. Unstable network conditions such as network jitter may result in message duplication. In this case, the consumer can perform idempotent processing for services that are sensitive to duplicated messages.
- Request parameters

Parameter	Type	Description
message	Message	The message that you want to consume.
context	ConsumeContext	The context of the consumption. For more information, see ConsumeContext in the code.

- Response parameters

Parameter	Description
Action	The result of the consumption. Consumption failed: An exception is thrown or Action.ReconsumeLater is returned. Action.ReconsumeLater indicates null. Consumption succeeded: Action.CommitMessage is returned.

4.1.5.6.4. Action

4.1.5.6.4.1. Overview

The result of the consumption that is returned for a message.

Enumeration

Value	Type	Description
CommitMessage	String	The message is consumed. The consumer continues to consume the next message.
ReconsumeLater	String	The consumer failed to consume the message. The consumer requests the RocketMQ broker to redeliver the failed message later and continues to consume another message.

4.1.5.6.4.2. valueOf

- Syntax: `public static Action valueOf(String name)`
- Description: Returns an enumerated constant of an enumerated type with the specified name. The specified string must exactly match the ID that is used to declare the enumerated constant. No extra space is allowed.
- Request parameters

Parameter	Type	Description
name	String	The name of the enumerated constant to return.

- Response parameters

Parameter	Description
Action	The enumerated constant with the specified name that is returned.

- Exceptions

Exception	Description
IllegalArgumentException	This exception is thrown because the enumerated type does not have an enumerated constant with the specified name.
NullPointerException	This exception is thrown because the value of the request parameter is null.

4.1.5.6.4.3. values

- Syntax: `public static Action[] values()`
- Description: Returns an array of constants in the declared order of constants of an enumerated type.
- Request parameters: none
- Response parameters

Parameter	Description
Action[]	The array of constants in the declared order of constants of the enumerated type that is returned.

4.1.5.6.5. PropertyValueConst

The constant that specifies the consumption mode.

Enumeration

Value	Type	Description
BROADCASTING	String	The broadcasting consumption mode.
CLUSTERING	String	The clustering consumption mode.

4.1.5.6.6. OrderConsumer

4.1.5.6.6.1. Overview

You can use a consumer instance of ordered messages to subscribe to messages in first in, first out (FIFO) order.

4.1.5.6.6.2. subscribe

- Syntax: void subscribe(String topic, String subExpression, MessageOrderListener listener)
- Description: Subscribes to ordered messages.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.
subExpression	String	The filter expression for subscribing to messages. The RocketMQ broker filters messages based on this expression. Only OR () operations are supported. Example 1: "tag1 tag2 tag3"; Example 2: When the subExpression parameter is set to null or *, all messages are subscribed to.
listener	MessageOrderListener	The message listener for callbacks.

- Response parameters: none

4.1.5.6.6.3. unsubscribe

- Syntax: void unsubscribe(String topic)
- Description: Cancels the subscription to a specific topic.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.

- Response parameters: none

4.1.5.6.6.4. `isStarted`

- Syntax: `public boolean isStarted()`
- Description: Checks whether an instance is enabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.6.6.5. `isClosed`

- Syntax: `public boolean isClosed()`
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.6.6.6. `start`

- Syntax: `public void start()`
- Description: Enables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.6.7. `shutdown`

- Syntax: `public void shutdown()`
- Description: Disables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.7. MessageOrderListener

4.1.5.6.7.1. Overview

A consumer registers an ordered message listener to subscribe to ordered messages.

Thread security requirements: A message listener can be concurrently used by multiple threads of a consumer. You must ensure the security of concurrency.

4.1.5.6.7.2. consume

- Syntax: `OrderAction consume(Message message, ConsumeOrderContext context)`
- Description: Consumes messages. This method is called by applications. Unstable network conditions such as network jitter may result in message duplication. In this case, the consumer can perform idempotent processing for services that are sensitive to duplicated messages.
- Request parameters

Parameter	Type	Description
message	Message	The message that you want to consume.
context	ConsumeOrderContext	The context of the consumption. For more information, see <code>ConsumeOrderContext</code> in the code.

- Response parameters

Parameter	Description
OrderAction	The result of the consumption. Consumption failed: An exception is thrown or <code>Action.Suspend</code> is returned. <code>Action.Suspend</code> indicates null. Consumption succeeded: <code>Action.Success</code> is returned.

4.1.5.6.8. OrderAction

4.1.5.6.8.1. Overview

The result of the consumption that is returned for an ordered message.

Enumeration

Value	Type	Description
Success	String	The message is consumed. The consumer continues to consume the next message.
Suspend	String	The consumer failed to consume the message. The current queue is suspended.

4.1.5.6.8.2. valueOf

- Syntax: `public static OrderAction valueOf(String name)`
- Description: Returns an enumerated constant of an enumerated type with the specified name. The specified string must exactly match the ID that is used to declare the enumerated constant. No extra space is allowed.
- Request parameters

Parameter	Type	Description
name	String	The name of the enumerated constant to return.

- Response parameters

Parameter	Description
OrderAction	The enumerated constant with the specified name that is returned.

- Exceptions

Exception	Description
IllegalArgumentException	This exception is thrown because the enumerated type does not have an enumerated constant with the specified name.
NullPointerException	This exception is thrown because the value of the request parameter is null.

4.1.5.6.8.3. values

- Syntax: `public static OrderAction[] values()`
- Description: Returns an array of constants in the declared order of constants of an enumerated type.
- Request parameters: none
- Response parameters

Parameter	Description
OrderAction[]	The array of constants in the declared order of constants of the enumerated type that is returned.

4.1.5.6.9. BatchConsumer

4.1.5.6.9.1. Overview

You can use a batch consumer instance to subscribe to multiple messages at a time.

4.1.5.6.9.2. subscribe

- Syntax: `void subscribe(String topic, String subExpression, BatchMessageListener listener)`
- Description: Subscribes to regular messages, transactional messages, scheduled messages, or delayed messages.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.
subExpression	String	The filter expression for subscribing to messages. The RocketMQ broker filters messages based on this expression. Only OR () operations are supported. Example 1: "tag1 tag2 tag3"; Example 2: When the subExpression parameter is set to null or *, all messages are subscribed to.
listener	BatchMessageListener	The message listener for callbacks.

- Response parameters: none

4.1.5.6.9.3. unsubscribe

- Syntax: `void unsubscribe(String topic)`
- Description: Cancels the subscription to a specific topic.
- Request parameters

Parameter	Type	Description
topic	String	The topic of the messages.

- Response parameters: none

4.1.5.6.9.4. isStarted

- Syntax: `public boolean isStarted()`
- Description: Checks whether an instance is enabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is enabled. A value of true indicates that the instance is enabled. A value of false indicates that the instance is not enabled.

4.1.5.6.9.5. isClosed

- Syntax: `public boolean isClosed()`
- Description: Checks whether an instance is disabled.
- Request parameters: none
- Response parameters

Type	Description
boolean	Indicates whether the instance is disabled. A value of true indicates that the instance is disabled. A value of false indicates that the instance is not disabled.

4.1.5.6.9.6. start

- Syntax: `public void start()`
- Description: Enables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.9.7. shutdown

- Syntax: `public void shutdown()`
- Description: Disables a consumer instance.
- Request parameters: none
- Response parameters: none

4.1.5.6.10. BatchMessageListener

4.1.5.6.10.1. Overview

A consumer registers a batch message listener to consume multiple messages at a time.

Thread security requirements: A message listener can be concurrently used by multiple threads of a consumer. You must ensure the security of concurrency.

4.1.5.6.10.2. consume

- Syntax: `Action consume(Message message, ConsumeContext context)`
- Description: Consumes multiple messages at a time. This method is called by applications. Unstable network conditions such as network jitter may result in message duplication. In this case, the consumer can perform idempotent processing for services that are sensitive to duplicated messages.
- Request parameters

Parameter	Type	Description
message	Message	The messages that you want to consume.
context	ConsumeContext	The context of the consumption. For more information, see <code>ConsumeContext</code> in the code.

- Response parameters

Parameter	Description
Action	The result of the consumption. Consumption failed: An exception is thrown or Action.ReconsumeLater is returned. Action.ReconsumeLater indicates null. Consumption succeeded: Action.CommitMessage is returned.