

ALIBABA CLOUD

阿里云

专有云企业版

消息队列 MQ
RocketMQ开发指南

产品版本：V3.14.0

文档版本：20210425

 阿里云

法律声明

阿里云提醒您,在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.通过ASAPI调用云产品SDK	12
1.1. ASAPI概述	12
1.2. 准备工作	13
1.2.1. 登录API与工具控制台	13
1.2.2. 查看API信息	13
1.2.3. 生成ASAPI的Endpoint	14
1.2.4. 获取ASAPI的SDK包	16
1.2.5. 获取AccessKey	16
1.2.6. 获取Region ID	18
1.3. 公共参数	18
1.4. SDK说明	19
1.4.1. Java SDK调用	19
1.4.2. Python SDK调用	24
1.4.3. Node SDK调用	26
1.4.4. POP SDK调用	28
1.4.4.1. Java版	28
1.4.4.2. Go版	31
1.4.4.3. Python版	33
1.5. HTTP调用方式	35
1.5.1. 概述	35
1.5.2. 请求结构	35
1.5.3. 签名机制	36
1.5.4. 返回结果	40
2.API参考	41
2.1. 产品更新说明	41
2.2. RocketMQ Open API简介	42

2.2.1. 面向用户和场景	42
2.2.2. 适用场景和局限性	42
2.2.3. Open API基本原理	43
2.3. RocketMQ Open API接入指南	43
2.3.1. 公共参数设置	43
2.3.2. 公共返回值解析	44
2.4. RocketMQ Open API公共术语	45
2.4.1. 账号安全类	45
2.4.1.1. AccessKey	45
2.4.1.2. SecretKey	45
2.4.2. RocketMQ资源管控类	45
2.4.2.1. Topic	45
2.4.2.2. Group ID (GID)	45
2.5. RocketMQ Open API错误码列表	45
2.6. API详细说明	46
2.6.1. Region管理	46
2.6.1.1. 获取regionId列表 (/json/region/list)	46
2.6.2. 集群管理	47
2.6.2.1. 获取集群信息 (/json/cluster/list)	47
2.6.3. 实例管理	48
2.6.3.1. 实例网络类型切换 (/json/vpcTunnel/tunnelSwitch)	48
2.6.3.2. 获取实例基本信息 (/json/instance/baseInfo)	49
2.6.3.3. 创建实例 (/json/instance/create)	50
2.6.3.4. 删除实例 (/json/instance/delete)	51
2.6.3.5. 获取实例列表 (/json/instance/list)	52
2.6.3.6. 更新实例名称和描述 (json/instance/update)	52
2.6.4. Topic管理	53
2.6.4.1. 创建Topic (/json/topic/create)	53

2.6.4.2. 删除Topic (/json/topic/delete)	55
2.6.4.3. 查询Topic (/json/topic/get)	55
2.6.4.4. 查询Topic列表 (/json/topic/listInPage)	56
2.6.4.5. 配置Topic读写权限 (json/topic/update)	56
2.6.4.6. 查询Topic名字 (/json/topic/listNames)	57
2.6.4.7. 查询Topic的订阅关系 (json/topic/subDetail)	57
2.6.5. Group管理	58
2.6.5.1. 创建Group (json/group/create)	59
2.6.5.2. 删除Group (json/group/delete)	60
2.6.5.3. 获取Group信息列表 (json/group/list)	60
2.6.5.4. 查询Group订阅关系 (json/group/subDetail)	61
2.6.5.5. 配置订阅权限 (json/group/consumerUpdate)	62
2.6.6. 消费管理	63
2.6.6.1. 查询消费堆积 (/json/consumer/accumulate)	63
2.6.6.2. 查询消费状态 (/json/consumer/status)	64
2.6.6.3. 查询消费端连接信息 (/json/consumer/getConnection)	68
2.6.6.4. 查询消息可重置的时间范围 (/json/consumer/timespan)	69
2.6.6.5. 重置消费位点 (/json/consumer/resetOffset)	70
2.6.7. 消息查询	71
2.6.7.1. 查询消息投递状态 (/json/message/trace)	71
2.6.7.2. 根据Message ID查询消息 (/json/message/getById)	72
2.6.7.3. 根据Message Key查询消息 (/json/message/getByKey)	74
2.6.7.4. 根据Topic查询消息 (/json/message/getByPagedTopic)	75
2.6.8. 发布订阅统计	77
2.6.8.1. Topic写入统计 (/json/trend/topicInputTps)	77
2.6.8.2. Group ID投递统计 (/json/trend/groupOutputTps)	78
2.6.9. 授权管理	80
2.6.9.1. 创建主账号授权 (/json/empower/create)	80

2.6.9.2. 创建子账号授权 (/json/empower/createBySub)	80
2.6.9.3. 删除授权 (/json/empower/delete)	81
2.6.9.4. 查询授权列表 (/json/empower/list)	82
2.6.10. 死信队列管理	82
2.6.10.1. 按Message ID查询死信消息 (/json/message/getByIdIn...)	82
2.6.10.2. 按Group ID查询死信消息 (/json/message/getByPaged..)	83
2.6.10.3. 重发死信消息 (/json/message/resendByIdInDLQ)	85
3.SDK参考	87
3.1. 概述	87
3.2. 消息模型	87
3.3. 产品更新说明	87
3.4. 传输协议与字符编码	89
3.5. 常用类及其方法	89
3.5.1. 常量类PropertyKeyConst	89
3.5.2. 工厂类ONSFactory	90
3.5.2.1. createConsumer	90
3.5.2.2. createBatchConsumer	91
3.5.2.3. createOrderedConsumer	91
3.5.2.4. createOrderProducer	92
3.5.2.5. createProducer	92
3.5.2.6. createTransactionProducer	93
3.5.3. 消息类Message	93
3.5.3.1. 消息类Message概述	93
3.5.3.2. Message默认构造方法	93
3.5.3.3. Message有参构造方法 (一)	93
3.5.3.4. Message有参构造方法 (二)	94
3.5.3.5. putUserProperties	94
3.5.3.6. getUserProperties	94

3.5.3.7. getTopic	95
3.5.3.8. setTopic	95
3.5.3.9. getTag	95
3.5.3.10. setTag	95
3.5.3.11. getKey	96
3.5.3.12. setKey	96
3.5.3.13. getMsgID	96
3.5.3.14. setMsgID	96
3.5.3.15. getReconsumeTimes	97
3.5.3.16. setReconsumeTimes	97
3.5.3.17. getBornTimestamp	97
3.5.3.18. setBornTimestamp	97
3.5.3.19. getBornHost	98
3.5.3.20. setBornHost	98
3.5.3.21. getStartDeliverTime	98
3.5.3.22. setStartDeliverTime	98
3.5.3.23. getShardingKey	99
3.5.3.24. setShardingKey	99
3.5.4. 异常类ONSClientException	99
3.5.4.1. ONSClientException默认构造方法	99
3.5.4.2. ONSClientException有参构造方法（一）	99
3.5.4.3. ONSClientException有参构造方法（二）	100
3.5.4.4. ONSClientException有参构造方法（三）	100
3.5.5. 发送消息接口	100
3.5.5.1. 发送消息接口概述	100
3.5.5.2. 发送结果SendResult	100
3.5.5.2.1. SendResult默认构造方法	100
3.5.5.2.2. getMsgId	101

3.5.5.2.3. getTopic	101
3.5.5.2.4. toString	101
3.5.5.3. 生产者实例Producer	101
3.5.5.3.1. 生产者实例Producer概述	101
3.5.5.3.2. send	102
3.5.5.3.3. sendOneway	102
3.5.5.3.4. sendAsync	102
3.5.5.3.5. isStarted	102
3.5.5.3.6. isClosed	103
3.5.5.3.7. start	103
3.5.5.3.8. shutdown	103
3.5.5.4. 分布式事务消息生产者实例 TransactionProducer	103
3.5.5.4.1. send	103
3.5.5.4.2. isStarted	104
3.5.5.4.3. isClosed	104
3.5.5.4.4. start	104
3.5.5.4.5. shutdown	104
3.5.5.4.6. 本地事务执行器LocalTransactionExecuter	105
3.5.5.4.6.1. execute	105
3.5.5.4.7. 分布式事务消息状态TransactionStatus	105
3.5.5.5. 顺序消息生产者实例OrderProducer	105
3.5.5.5.1. send	105
3.5.5.5.2. isStarted	106
3.5.5.5.3. isClosed	106
3.5.5.5.4. start	106
3.5.5.5.5. shutdown	106
3.5.6. 订阅消息接口	107
3.5.6.1. 订阅消息接口概述	107

3.5.6.2. 消费者实例Consumer	107
3.5.6.2.1. 消费者实例Consumer概述	107
3.5.6.2.2. subscribe	107
3.5.6.2.3. unsubscribe	108
3.5.6.2.4. isStarted	108
3.5.6.2.5. isClosed	108
3.5.6.2.6. start	108
3.5.6.2.7. shutdown	108
3.5.6.3. 消息监听器MessageListener	109
3.5.6.3.1. 消费监听器MessageListener概述	109
3.5.6.3.2. consume	109
3.5.6.4. 消息消费结果Action	109
3.5.6.4.1. 消息消费结果Action概述	109
3.5.6.4.2. valueOf	110
3.5.6.4.3. values	110
3.5.6.5. 消费模式PropertyValueConst	110
3.5.6.6. 顺序消费者实例OrderConsumer	111
3.5.6.6.1. 顺序消费者实例OrderConsumer概述	111
3.5.6.6.2. subscribe	111
3.5.6.6.3. unsubscribe	111
3.5.6.6.4. isStarted	111
3.5.6.6.5. isClosed	112
3.5.6.6.6. start	112
3.5.6.6.7. shutdown	112
3.5.6.7. 消息监听器MessageOrderListener	112
3.5.6.7.1. 消息监听器MessageOrderListener概述	112
3.5.6.7.2. consume	112
3.5.6.8. 顺序消息消费结果OrderAction	113

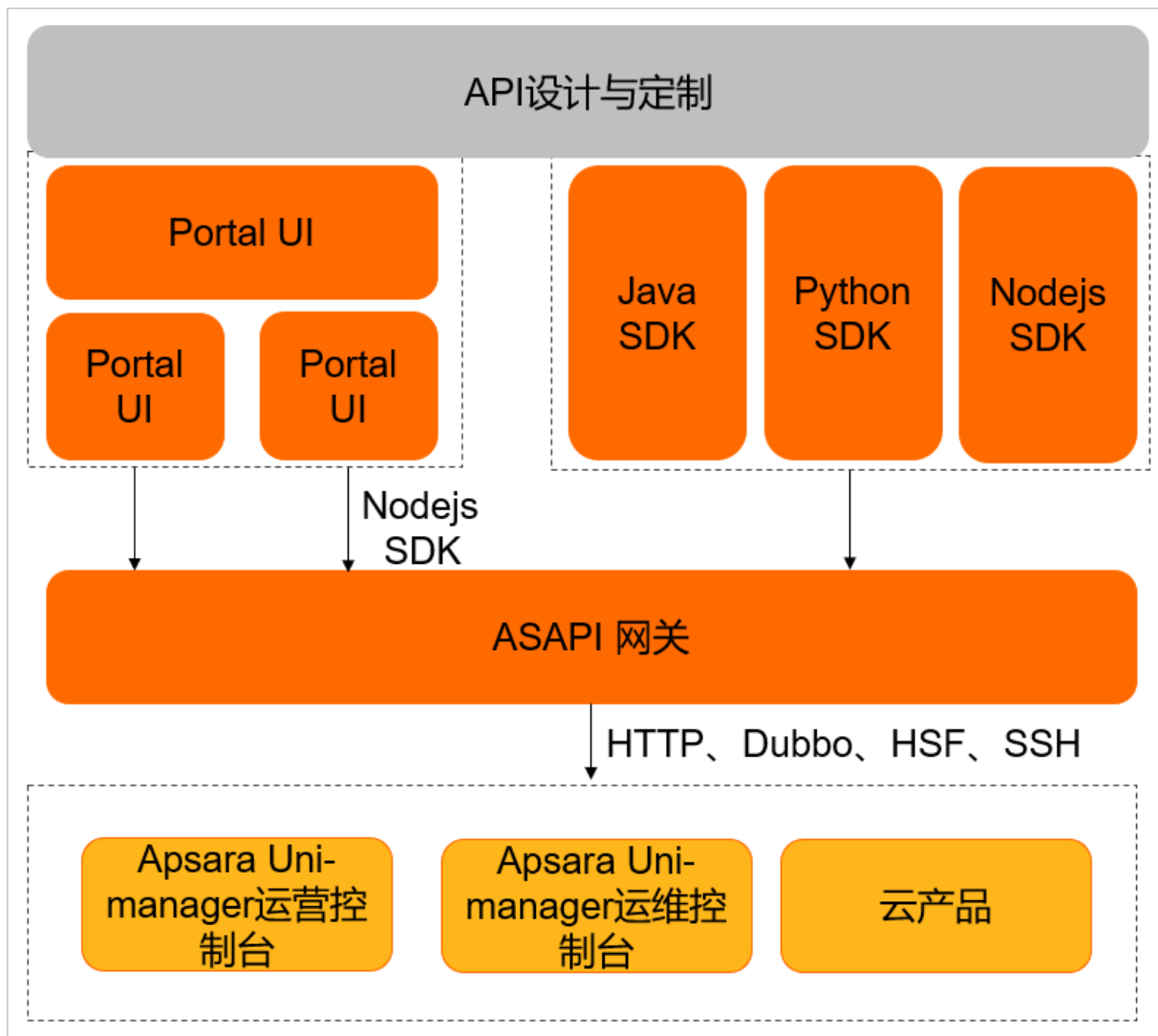
3.5.6.8.1. 顺序消息消费结果OrderAction概述	113
3.5.6.8.2. valueOf	113
3.5.6.8.3. values	114
3.5.6.9. 批量消费者实例BatchConsumer	114
3.5.6.9.1. 批量消费者实例BatchConsumer概述	114
3.5.6.9.2. subscribe	114
3.5.6.9.3. unsubscribe	114
3.5.6.9.4. isStarted	115
3.5.6.9.5. isClosed	115
3.5.6.9.6. start	115
3.5.6.9.7. shutdown	115
3.5.6.10. 批量消费消息监听器BatchMessageListener	116
3.5.6.10.1. 批量消费消息监听器BatchMessageListener概述	116
3.5.6.10.2. consume	116

1.通过ASAPI调用云产品SDK

1.1. ASAPI概述

本文主要从功能和架构等方面对专有云ASAPI（Apsara Stack API）系统进行简单的介绍。

ASAPI是阿里云专有云的统一入口网关，是阿里云混合云管理平台（Apsara Uni-manager）的核心组成部分。ASAPI定位于专有云API的统一管理和调用，承接了ECS、VPC等绝大多数专有云产品的接入，对外呈现统一的调用方式（统一SDK方式，可调用所有支持的专有云产品）。



ASAPI提供混合云管理相关的运营与运维API接口，包括：

- 组织、用户、角色、资源集、登录策略等企业管理API（相关功能由Apsara Uni-manager运营控制台提供）。
- 配额、计量、规格、密码策略等运营管理API（相关功能由Apsara Uni-manager运营控制台提供）。
- 云产品相关创建、删除、查询、修改等资源管理API（例如创建ECS、创建VPC，相关功能由ECS、VPC等云产品提供）。
- 云产品监控指标查询API（相关功能由Apsara Uni-manager运营控制台提供）。
- 监控告警、库存管理、任务管理、物理平台等运维相关API（相关功能由Apsara Uni-manager运维控制台提供）。

提供)。

上述API可以查询相应的开发指南，例如，组织用户相关API可查询*Apsara Uni-manager运营控制台开发指南*，创建ECS的API则需要查询*云服务器ECS开发指南*。

 **说明** 云产品的开发指南中，如果包含“通过ASAPI调用云产品SDK”，则说明专有云产品已经接入ASAPI，可参照本文进行API调用。

1.2. 准备工作

1.2.1. 登录API与工具控制台

您可以通过Apsara Uni-manager运营控制台登录API与工具控制台。


前提条件

- 登录Apsara Uni-manager运营控制台前，确认您已从部署人员处获取Apsara Uni-manager运营控制台的服务域名地址。
- 推荐使用Chrome浏览器。

操作步骤

1. 在浏览器地址栏中，输入Apsara Uni-manager运营控制台的访问地址，按回车键。
2. 输入正确的用户名及密码。

请向运营管理员获取登录控制台的用户名和密码。

 **说明** 首次登录Apsara Uni-manager运营控制台时，需要修改登录用户名的密码，请按照提示完成密码修改。为提高安全性，密码长度必须为8~20位，且至少包含以下两种类型：

- 英文大写或小写字母（A~Z、a~z）
- 阿拉伯数字（0~9）
- 特殊符号（感叹号（!）、at（@）、井号（#）、美元符号（\$）、百分号（%）等）

3. 单击**登录**。
4. 在页面顶部的菜单栏中，选择**产品 > 其他 > API与工具**。

1.2.2. 查看API信息

您需要获取产品名称、API名称和API版本，使用SDK时需要填写这些信息。

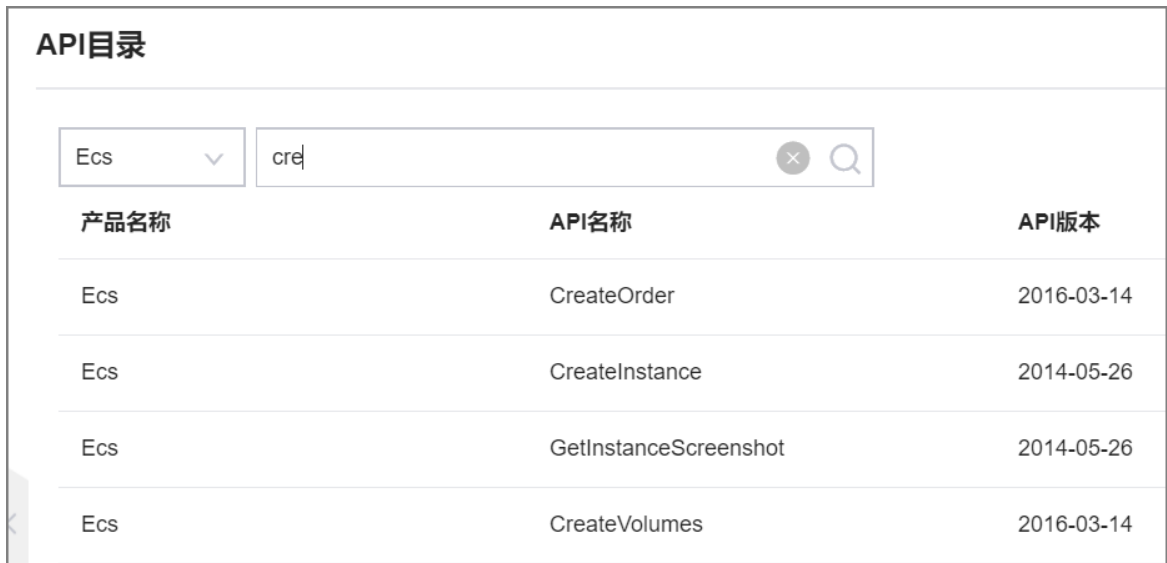
背景信息

API目录中的API仅供调用参考，具体API的使用方式需要参见云产品的开发指南。如果发生不一致情况，以开发指南支持情况为准。

操作步骤

1. [登录API与工具控制台](#)。
2. 在左侧导航栏中选择**API目录**。

3. 在API目录页面，在下拉列表框中选择对应的产品，输入API名称的关键词进行搜索。



4. 查看并记录对应的产品名称、API名称和API版本。

1.2.3. 生成ASAPI的Endpoint

本文将为您介绍三种获取Endpoint的方法。

方法一：通过服务注册变量获取Endpoint

1. 登录天基。
 - i. 登录Apsara Uni-manager运维控制台。
 - ii. 在顶部菜单栏单击运维。
 - iii. 在左侧导航栏中选择产品运维管理>产品列表。
 - iv. 单击天基。
2. 在左侧导航栏中单击报表。

3. 在全部报表页面搜索 服务注册变量 ，单击报表名称。



4. 在服务注册变量页面，单击Service旁的 ≡ 图标，搜索 asapi 。



5. 在asapi的Service Registration列中，单击鼠标右键，选择显示更多。

在详情页面，查看的asapi.public.endpoint值。

6. 按照以下格式生成ASAPI的Endpoint。

```
https://<asapi.public.endpoint>/asapi/v3
```

方法二：通过资源申请报表获取Endpoint

1. 登录天基。

- i. 登录Apsara Uni-manager运维控制台。
- ii. 在顶部菜单栏单击运维。
- iii. 在左侧导航栏中单击产品运维管理>产品列表。
- iv. 单击天基。

2. 进入集群运维。

- i. 在左侧导航栏中选择运维>集群运维。
- ii. 在集群文本框中输入 ascm 。
- iii. 单击操作栏中的运维。

3. 获取domain值。
 - i. 单击集群资源页签。
 - ii. 在名称文本框中输入 `asapi_dns_public` 。
 - iii. 单击申请参数栏中的详情。

在申请参数页面查看domain值。

```
{
  "domain": "public.asapi.██████████.com",
  "ha_apply_type": "active-active",
  "name": "asapi_dns_public",
  "vip": "██████████"
}
```

4. 按照以下格式生成ASAPI的Endpoint。

```
https://<domain>/asapi/v3
```

方法三：自行拼装Endpoint

您还可以按照以下格式自行拼装Endpoint。

```
https://public.asapi.<region>.<internet-domain>/asapi/v3
```

② 说明

1. region: 替换为真实的地域ID值，获取方式请参见[获取Region ID](#)。
2. internet-domain: 替换为专有云的根域名（特别提示是internet，不是intranet），可咨询专有云运维管理员。

假设region示例值为 `cn-****-****-d01`，internet-domain示例值为 `inter.****.****.com`。

完整的Endpoint示例值如下：

```
https://public.asapi.cn-****-****-d01.inter.****.****.com/asapi/v3
```

1.2.4. 获取ASAPI的SDK包

专有云环境中使用ASAPI作为所有接入ASAPI产品的统一入口，并提供SDK进行产品侧接口的调用支持。

请从[专有云文档官网](#)获取对应版本ASAPI的SDK包。

1.2.5. 获取AccessKey

访问身份验证中通过使用AccessKey ID和AccessKey Secret对称加密的方法来验证某个请求的发送者身份。AccessKey ID用于标示用户，AccessKey Secret是用户用于加密签名字符串。本章节将为您描述如何获取AccessKey。

前提条件

只有运营管理员和一级组织管理员可以获取组织AccessKey。

背景信息

推荐使用个人账号AccessKey进行调用Apsara Uni-manager运营控制台及云产品相关接口。如果使用个人账号AccessKey需要在Header中添加以下限制性参数，否则可能提示权限不足：

参数名称	描述
x-acs-regionid	环境Region如：cn-hangzhou-*。
x-acs-organizationid	Apsara Uni-manager运营控制台中对应的组织ID。获取方法可参见Apsara Uni-manager运营控制台开发指南中的GetOrganizationList文档。
x-acs-resourcegroupid	Apsara Uni-manager运营控制台中对应的资源集ID。获取方法可参见Apsara Uni-manager运营控制台开发指南中的ListResourceGroup文档。
x-acs-instanceid	进行操作的实例ID，例如，ECS的实例ID可以从ECS列表中查询到。

警告 个人账号AccessKey是由Apsara Uni-manager运营控制台权限体系管控的受限AccessKey，组织AccessKey相对权限较大，需要管理员确认操作的安全性。

获取个人账号AccessKey

获取个人账号AccessKey的方法如下：

1. 登录Apsara Uni-manager运营控制台。
2. 在系统界面右上角，单击当前登录用户的头像，选择**个人信息**。
3. 在**阿里云AccessKey**区域，您可以查看个人账户的AccessKey信息。



说明 Accesskey ID和AccessKey Secret是您访问云资源时的密钥，具有该账号完整的权限，请您妥善保管。

获取组织AccessKey

获取组织AccessKey的方法如下：

1. 管理员登录Apsara Uni-manager运营控制台。
2. 在页面顶部菜单栏上，单击**企业**。
3. 在**企业**页面的左侧导航栏中，单击**组织管理**。
4. 在组织结构中，单击目标组织名称。
5. 在右侧区域中，单击**获取accesskey**。

6. 在弹出的对话框中，查看组织Accesskey信息。

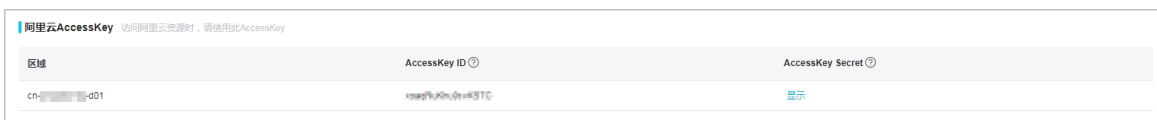
说明 一级组织的AccessKey为系统自动分配，下级组织使用一级组织的AccessKey。

1.2.6. 获取Region ID

Nodejs调用时需要设置Region ID。请从Apsara Uni-manager运营控制台获取Region ID。

操作步骤

1. 管理员登录Apsara Uni-manager运营控制台。
2. 在系统界面右上角，单击当前登录用户的头像，选择个人信息。
3. 在阿里云AccessKey区域，您可以查看区域（Region）ID。



1.3. 公共参数

公共参数是所有接口调用需要的参数。

公共请求参数

以下是调用ASAPI服务的公共请求参数：

名称	类型	是否必需	描述
Action	String	是	产品注册在ASAPI的API名称，可通过 查看API信息 查看。
Product	String	是	产品名称。
Version	String	是	产品注册在ASAPI的API版本，可通过 查看API信息 查看。
RegionId	String	是	Region ID。可通过 获取Region ID 查看。
AccessKeyId	String	是	访问服务使用的密钥ID，可通过 获取AccessKey 查看。

名称	类型	是否必需	描述
Signature	String	是	您的签名。取值请参见签名机制。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1，HMAC-SHA256。
SignatureVersion	String	是	签名算法版本。取值：1.0或2.0。
SignatureNonce	String	是	签名唯一随机数。用于防止网络重放攻击，建议您每一次请求都使用不同的随机数。
Timestamp	String	是	请求的时间戳。按照ISO8601标准表示，并需要使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。示例：2018-01-01T12:00:00Z表示北京时间2018年01月01日20点00分00秒。
Format	String	是	返回参数的语言类型。取值范围：JSON。

说明

其中 `Signature`、`SignatureMethod`、`SignatureVersion`、`SignatureNonce`、`Timestamp`、`Format` 六个参数已经封装在SDK内部，不需要提供。

1.4. SDK说明

1.4.1. Java SDK调用

本文介绍如何获取Java SDK并开始调用。

前提条件

- 使用阿里云专有云Java SDK，您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。参见[获取AccessKey](#)在Apsara Uni-manager运营控制台上获取和查看您的AccessKey。

- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。
- 阿里云专有云Java SDK需要1.8及以上版本的JDK。

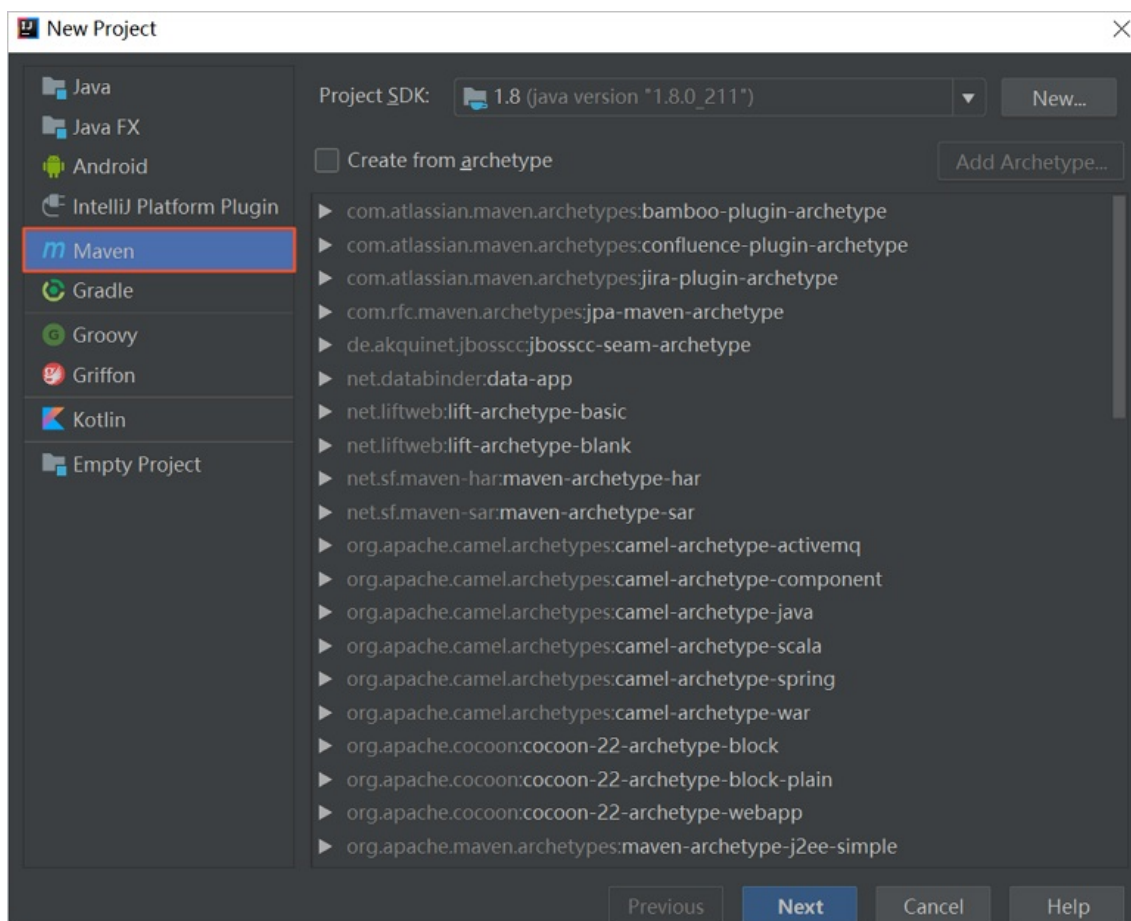
背景信息

专有云帮助中心为您提供了ASAPI Java SDK的Maven项目依赖及jar包，您可以编写代码调用ASAPI Java SDK来实现访问专有云网关服务ASAPI。

本文示例中，客户端为Windows10 64位操作系统，使用的Java开发工具为IntelliJ IDEA。

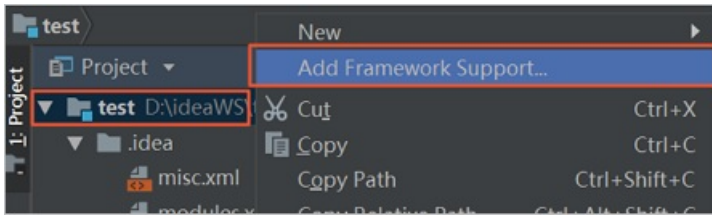
操作步骤

1. 通过以下任一方式在IDEA中配置Maven项目管理工具。
 - 使用IDEA中集成的Maven项目管理工具。
 - 访问Maven官方下载页面（[Download Apache Maven](#)）下载对应操作系统的Maven工具，手动配置Maven工具。
2. 从共享资源中获取ASAPI Java SDK的Maven项目依赖，获取方式请参见[获取ASAPI的SDK包](#)。
3. 通过以下任一方式创建Maven项目。
 - 方式一：在IDEA中添加一个Maven项目。

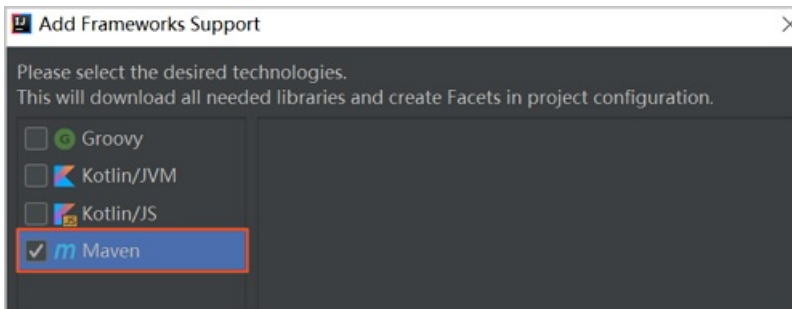


- 方式二：将已有的项目转换为Maven项目。

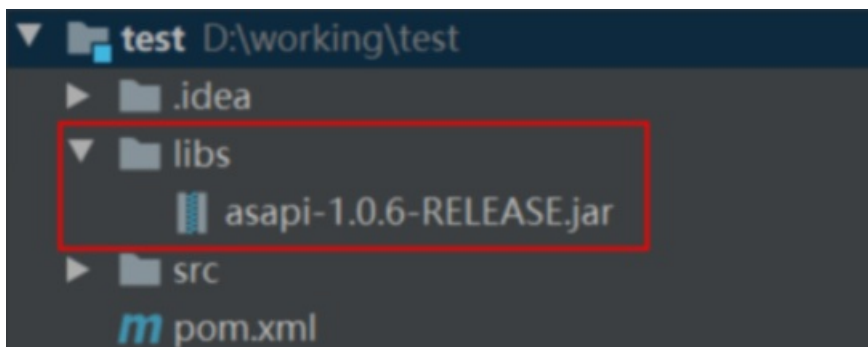
a. 右键单击要转换的项目，并选择Add Framework Support...。



b. 选择Maven，并单击OK。



4. 在项目根目录下创建libs包，将ASAPIJava SDK包放入到libs下，如下图。



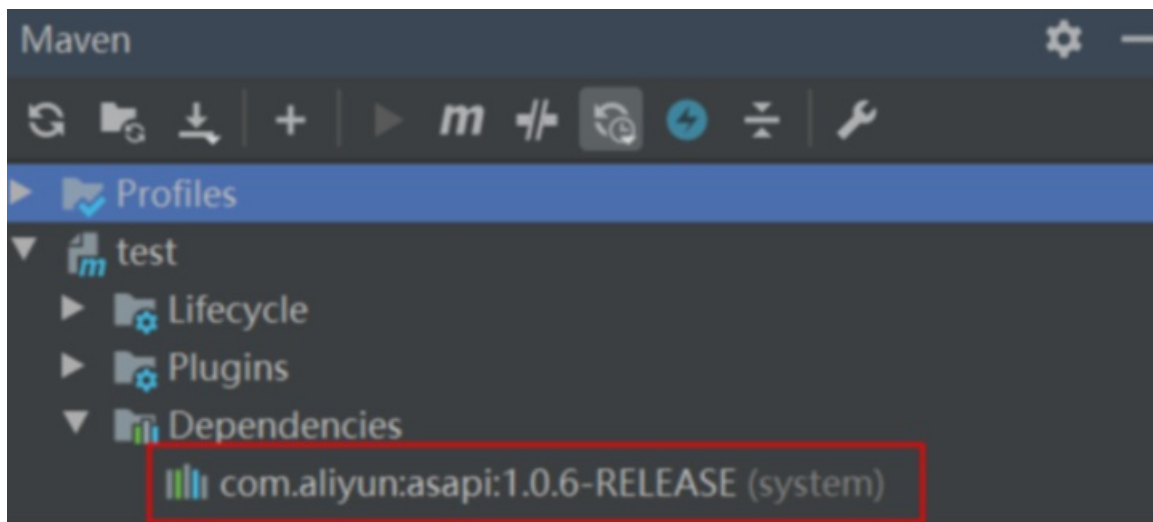
5. 在项目目录下的pom.xml中，添加ASAPIJava SDK依赖。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>asapi</artifactId>
    <version>1.0.6-RELEASE</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/libs/asapi-1.0.6-RELEASE.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>alicrypto-java-aliyun</artifactId>
    <version>1.0.4</version>
    <systemPath>${project.basedir}/libs/alicrypto-java-aliyun-1.0.4.jar</systemPath>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>gmsse</artifactId>
    <version>1.1.0</version>
    <systemPath>${project.basedir}/libs/gmsse-1.1.0.jar</systemPath>
  </dependency>
</dependencies>
```

说明 代码中的参数 1.0.6-RELEASE 需要根据实际SDK包版本号进行编写。

6. 查看Idea右侧Maven栏。

如图所示，表示已成功导入依赖。



使用Java SDK

目前支持使用GET或POST中任意一种方式调用ASAPI平台的API。调用示例如下：

```
// 1. 创建ASClient连接
ASClient client = new ASClient();
// 2. 设置身份标识,主要是标识调用来源,无实际作用,可随意设置,必填项
client.setSdkSource("<调用身份标识>");
Map<String, Object> parameters = new HashMap<String, Object>();
// 3. 设置授权相关参数 (与环境相关)
// 授权账号的AccessKey ID
parameters.put("AccessKeyId", "<AccessKey ID>");
// 授权账号的AccessKey Secret
parameters.put("AccessKeySecret", "<AccessKey Secret>");
// 地域Region
parameters.put("RegionId", "<your RegionID>");
// 4. 设置调用的接口信息,Product/Action/Version
parameters.put("Product", "<产品名称>");
parameters.put("Action", "<接口名称>");
parameters.put("Version", "<接口版本>");
// 5. 接口业务参数设置
parameters.put("<业务参数>", "<参数值>");
// 6. 创建Headers,并设置签名加密方式,目前支持HMAC-SHA1/HMAC-SHA256两种,默认使用HMAC-SHA256
Map<String, String> headers = new HashMap<String, String>();
//headers.put("SignatureMethod", "HMAC-SHA1");
headers.put("x-acis-regionid", "<your-regionid>");
headers.put("x-acis-resourcegroupid", "<your-resourcegroupid>");
headers.put("x-acis-organizationid", "<your-organizationid>");
headers.put("x-acis-instanceid", "<your-instanceid>");
// 7. 发起调用,调用方式推荐使用doPost
// 设置连接超时
//client.setConnectTimeout(10000);
// 设置请求超时
// client.setSocketTimeout(10000);
String result = client.doPost(endpoint, headers, parameters);
System.out.println(result);
```

设置代理

访问代理环境时调用添加代理如下:

```
Cloud cloud = new Cloud();
// 设置IP
cloud.setConfig("proxy_host", "<IP>");
// 设置端口
cloud.setConfig("proxy_port", "<Port>");
// 1. 创建ASClient连接
ASClient client = new ASClient(cloud);
```

Header参数描述

名称	描述
x-acis-regionid	环境Region, 获取方式请参见 获取Region ID 。
x-acis-organizationid	Apsara Uni-manager运营控制台上的组织ID。

名称	描述
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acs-instanceid	进行操作的实例ID。

1.4.2. Python SDK调用

您可以使用ASAPI提供的Python SDK进行API调用。

环境准备

- 用阿里云专有云Python SDK，您需要一个授权账号以及一对AccessKey ID和AccessKey Secret。请参见[获取AccessKey](#)在Apsara Uni-manager运营控制台上获取和查看您的AccessKey。
- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。

安装ASAPI Python SDK包

1. 下载对应版本的SDK包。具体操作请参见[获取ASAPI的SDK包](#)。
2. 将SDK包中Python SDK的压缩包上传至需要运行Python API调用程序的目标主机中并解压，然后在SDK文件包所在目录下执行以下代码安装SDK包：

```
# Python 2.x执行
pip install aliyun-python2-sdk-asapi-2.4.7.tar.gz
# Python 3.x执行
pip3 install aliyun-python3-sdk-asapi-2.4.7.tar.gz
```

🔍 说明

代码中的 `2.4.7` 需要根据实际使用的SDK包版本号进行编写。

调用示例

目前支持GET和POST两种方式调用ASAPI平台的API。


```
# 1. 引入依赖
from aliyunsdkasapi.ASClient import ASClient
from aliyunsdkasapi.AsapiRequest import AsapiRequest

if __name__ == '__main__':
    # 2. 创建Request,Product产品名称,Version接口版本,Action接口名称,asapi-endpoint ASAPI服务访问域名
    req = AsapiRequest("<Product>", "<Version>", "<Action>", "<asapi-endpoint>")
    # 3. 业务信息
    req.add_body_params("<key>", "<value>")
    # 4. Header信息
    req.add_header("x-acs-resourcegroupid", "<your-resourcegroupid>");
    req.add_header("x-acs-organizationid", "<your-organizationid>");
    req.add_header("x-acs-regionid", "<your-regionid>");
    req.add_header("x-acs-instanceid", "<your-instanceid>");
    # 5. 设置请求方式
    req.set_method("POST")
    # 6. 用环境信息初始化ASClient timeout设置请求超时时间, cert_file设置证书文件, verify是否验证证书
    as_client = ASClient("<accesskeyid>", "<accesskeysecret>", "<regionid>", timeout=1000,
        cert_file=None,
        verify=False)
    # 设置调用源,主要是标识调用来源,无实际作用,可随意设置,必填项
    as_client.setSdkSource("<调用身份标识>")
    response = as_client.do_request(req)
    print(response)
```

Header参数描述

名称	描述
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acs-organizationid	Apsara Uni-manager运营控制台上的组织ID。
x-acs-regionid	环境Region, 获取方式请参见 获取Region ID 。
x-acs-instanceid	进行操作的实例ID。

设置代理

您需要按以下步骤设置代理：

1. 执行以下命令导入OS依赖。

```
import os
```

2. 执行以下命令设置代理。

```
# 调用开始前添加
os.environ['http_proxy'] = "http://x.x.x.x:xxxx"
os.environ['https_proxy'] = "https://x.x.x.x:xxxx"
```

1.4.3. Node SDK调用

本文介绍如何获取Node.js SDK并开始调用。

环境准备

- 使用阿里云专有云Node.js SDK，您需要获取授权账号、一对AccessKey ID和AccessKey Secret。具体操作请参见[获取AccessKey](#)。
- 获取ASAPI的Endpoint，具体操作请参见[生成ASAPI的Endpoint](#)。
- 获取Region ID，具体操作请参见[获取Region ID](#)。

添加依赖

执行以下命令安装Node.js SDK包。

```
tnpm install --save @ali/asapi-nodejs-sdk
```

使用Node.js SDK

以下展示了调用Node.js SDK的示例代码：

```
// 1. 引入调用SDK依赖
const Controller = require('egg').Controller;
const ASAPIClient = require('@ali/asapi-nodejs-sdk');
// 2. 设置配置
const cnofig = {
  API_GATEWAY: '<ASAPI的Endpoint>',
  regionId: '<云产品所在地域>',
  accessKeyId: '<授权账号的AK>',
  accessKeySecret: '<授权账号的SK>'
};
// 3. 初始化ASAPI客户端
const AsapiClient = new ASAPIClient(cnofig, this.ctx);
// 4. 配置请求参数
const params = {
  "Action": "<接口名称>",
  "Product": "<接口所属产品>",
  "Version": "<接口版本>",
  "regionId": "<云产品所在地域>",
  "<接口所需业务参数>": "<参数值>"
}
// 5. 添加headers
AsapiClient.addHeader('x-acs-caller-sdk-source', '<身份标识>')
AsapiClient.addHeader('x-acs-roleid', '<角色id>')
AsapiClient.addHeader('x-acs-userid', '<用户id>')
AsapiClient.addHeader('x-acs-organizationid', '<your-organizationid>')
AsapiClient.addHeader("x-acs-regionid", "<your-regionid>");
AsapiClient.addHeader("x-acs-resourcegroupid", "<your-resourcegroupid>");
AsapiClient.addHeader("x-acs-instanceid", "<your-instanceid>");
// 6. 发起调用并且获取返回结果
try {
  const res = await AsapiClient.doRequest(cnofig.API_GATEWAY, params, {
    method: 'POST', // 请求方式
    formatParams: false, // 是否格式化参数,首字母大写转化
    enableProxy: true, // 是否配置代理
    proxy: 'http://x.x.x.x:xxxx',
    timeout: 10000, // 超时时间
    // ca: ,// ca证书设置 详见 https://nodejs.org/api/https.html#https_https_request_options_callback
  })
  console.log(res)
} catch (e) {
  console.log(e)
}
```

设置忽略证书

以下展示了忽略证书的代码。

```
设置环境变量
NODE_TLS_REJECT_UNAUTHORIZED=0
```

Header参数描述

名称	描述
x-acs-regionid	环境Region, 获取方式请参见 获取Region ID 。
x-acs-organizationid	Apsara Uni-manager运营控制台上的组织ID。
x-acs-resourcegroupid	Apsara Uni-manager运营控制台上的资源集ID。
x-acs-instanceid	进行操作的实例ID。
x-acs-caller-sdk-source	输入自定义的调用方名称或者系统名称, 作为来源说明。
x-acs-roleid	角色的ID。
x-acs-userid	用户的ID。

1.4.4. POP SDK调用

1.4.4.1. Java版

通过POP Java SDK访问ASAPI服务, 需要了解该产品调用方式(Rpc或Roa), 目前ASAPI已经兼容了POP Java SDK这两种调用风格, 不过在设置访问ASAPI域名上有差异。

前提条件

- 通过 [阿里云开发工具包 \(SDK\)](#) 下载对应产品的安装依赖。
- HTTPS访问设置为忽略SSL证书验证。

背景信息

ASAPI仅支持HTTPS协议请求通信。

差异说明

Rpc与Roa访问域名的区别如下。

```
# Rpc方式访问域名
public.asapi.<global:region>.<global:internet-domain>/asapi/v3
# Roa方式访问域名
public.asapi.<global:region>.<global:internet-domain>/asapi/v3/roa
```

调用示例

RPC调用, 示例接口信息: 产品 (ECS)、接口 (DescribeInstances)、版本 (2014-05-26)。

调用示例如下:

```
package com.aliyuncs.client.pop.ecs;

import com.aliyuncs.DefaultAcscClient;
import com.aliyuncs.ecs.model.v20140526.DescribeInstancesRequest;
import com.aliyuncs.ecs.model.v20140526.DescribeInstancesResponse;
import com.aliyuncs.http.HttpClientConfig;
```

```
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;

import java.util.List;

public class DescribeInstancesTest {
    public static void main(String[] args) {
        // ASAPI服务访问地址
        String endpoint = "public.asapi.<global:region>.<global:internet-domain>/asapi/v3";
        // 当前环境所在ECS的地域
        String region = "<global:region>";
        // 产品名称
        String product = "ECS";
        // 访问密钥
        String ak = "<accessKeyId>";
        String sk = "<accessKeySecret>";
        try {
            DefaultProfile clientProfile = DefaultProfile.getProfile(region, ak, sk);
            // 忽略SSL
            HttpClientConfig clientConfig = HttpClientConfig.getDefault();
            clientConfig.setIgnoreSSLCerts(true);
            // 设置连接超时
            clientConfig.setConnectionTimeoutMillis(3000L);
            // 设置读取超时
            clientConfig.setReadTimeoutMillis(10000L);
            clientProfile.setHttpClientConfig(clientConfig);
            // 创建链连接
            DefaultAcsClient acsClient = new DefaultAcsClient(clientProfile);
            // 创建请求对象
            DescribeInstancesRequest request = new DescribeInstancesRequest();
            // 设置请求服务地址
            request.setSysEndpoint(endpoint);
            // 设置LocationProduct
            request.setSysLocationProduct(product);
            request.setSysRegionId(region);
            // 设置请求协议HTTPS
            request.setSysProtocol(ProtocolType.HTTPS);
            DescribeInstancesResponse result = acsClient.getAcsResponse(request);
            List<DescribeInstancesResponse.Instance> instances = result.getInstances();
            for (DescribeInstancesResponse.Instance instance: instances) {
                System.out.println(instance.getInstanceId());
                System.out.println(instance.getInstanceName());
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Roai调用，示例接口信息：产品（CR）接口（GetNamespaceList）版本（2016-06-07）。

调用示例如下：

```
package com.aliyuncs.client.pop.cr;

import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpClientConfig;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.http.ProtocolType;
import com.aliyuncs.profile.DefaultProfile;

public class GetNamespaceListTest {
    public static void main(String[] args) {
        // ASAPI服务访问地址
        String endpoint = "public.asapi.<global:region>.<global:internet-domain>/asapi/v3/roa";
        // 当前环境所在ECS的地域
        String region = "<global:region>";
        // 产品名称
        String product = "cr";
        // 访问密钥
        String ak = "<accessKeyId>";
        String sk = "<accessKeySecret>";
        DefaultProfile profile = DefaultProfile.getProfile(region, ak, sk);
        // 忽略SSL
        HttpClientConfig clientConfig = HttpClientConfig.getDefault();
        clientConfig.setIgnoreSSLCerts(true);
        // 设置连接超时
        clientConfig.setConnectionTimeoutMillis(3000L);
        // 设置读取超时
        clientConfig.setReadTimeoutMillis(10000L);
        profile.setHttpClientConfig(clientConfig);
        // 创建连接Client
        IAcsClient client = new DefaultAcsClient(profile);
        // 创建请求对象
        CommonRequest request = new CommonRequest();
        // 设置HTTPS访问
        request.setSysProtocol(ProtocolType.HTTPS);
        // 设置请求方式
        request.setSysMethod(MethodType.GET);
        // 设置ASAPI访问地址
        request.setSysDomain(endpoint);
        // 设置访问接口版本
        request.setSysVersion("2016-06-07");
        // 设置uriPattern
        request.setSysUriPattern("/namespace");
        // 设置LocationProduct
        request.setSysLocationProduct(product);

        request.putHeadParameter("Content-Type", "application/json");
        String requestBody = "" +
            "{}";
    }
}
```

```
request.setHttpContent(requestBody.getBytes(), "utf-8", FormatType.JSON);
try {
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

注意

设置HTTPS访问，一定要设置忽略SSL证书验证。

1.4.4.2. Go版

本文为您介绍POP Go SDK调用的方式。

前提条件

通过[阿里云开发工具包（SDK）](#)下载对应产品的安装依赖。

背景信息

目前仅支持通过Rpc方式调用API。

调用示例

调用示例接口信息：产品（ECS）、接口（DescribeInstances）、版本（2014-05-26）。

示例代码如下：

```
package main

import (
    "fmt"
    "time"

    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/auth/credentials"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/responses"
)

func getSdkConfig() *sdk.Config {
    return sdk.NewConfig().
        WithTimeout(time.Duration(30) * time.Second).
        WithEnableAsync(true).
        WithGoRoutinePoolSize(100).
        WithMaxTaskQueueSize(10000).
        WithDebug(false).
        WithScheme("HTTP")
}

func main(){
    ak := "<授权账号的AccessKey ID>"
    sk := "<授权账号的AccessKey Secret>"
    regionId := "<ECS所在地域>"
    endpoint := "<ASAPI的Endpoint>" // No network protocol
    request := requests.NewCommonRequest()
    request.SetReadTimeout(10 * time.Second) // Set request ReadTimeout to 10 second.
    request.SetConnectTimeout(5 * time.Second) // Set request ConnectTimeout to 5 second.
    request.Method = requests.GET
    request.Domain = endpoint // Location Service will not be enabled if the host is specified. For example, s
    ervice with a Certification type-Bearer Token should be specified
    request.Version = "2014-05-26" // Specify product version
    request.Scheme = "https" // Set request scheme. Default: http
    request.ApiName = "DescribeInstances"
    request.QueryParams["Product"] = "Ecs" // Specify product
    request.TransToAcsRequest()
    client, err := sdk.NewClientWithOptions(regionId, getSdkConfig(), credentials.NewAccessKeyCredential(ak, s
    k))
    // When the request is not set, the client settings are used.
    //client.SetReadTimeout(10 * time.Second) // Set client ReadTimeout to 10 second.
    //client.SetConnectTimeout(5 * time.Second) // Set client ConnectTimeout to 5 second.
    resp := responses.BaseResponse{}
    err = client.DoAction(request, &resp)
    if err != nil {
        panic(err)
    }
    fmt.Print(endpoint)
}
```


设置代理

您可以通过以下两种方式设置代理。

- 通过环境变量设置。

环境变量HTTP_PROXY、HTTPS_PROXY或者NO_PROXY。

设置代码如下：

```
import (
    "os"
)
func main(){
    os.Setenv("HTTP_PROXY", "http://x.x.x.x:xxxx") // 设置 HTTP_PROXY 环境变量
    os.Setenv("HTTPS_PROXY", "https://x.x.x.x:xxxx") // 设置 HTTPS_PROXY 环境变量
}
```

- 通过客户端设置。

设置代码如下：

```
// 客户端设置代理优先级比环境变量高
client.SetHttpProxy("http://x.x.x.x:xxxx") // 设置http代理
client.GetHttpProxy() // 获取http代理

client.SetHttpsProxy("https://x.x.x.x:xxxx") // 设置https代理
client.GetHttpsProxy() // 获取https代理

client.SetNoProxy("127.0.0.1,localhost") // 设置代理白名单
client.GetNoProxy() // 获取代理白名单
```

1.4.4.3. Python版

本文为您介绍POP Python SDK调用的方式。

前提条件

通过[阿里云开发工具包（SDK）](#)下载对应产品的安装依赖。

背景信息

目前仅支持通过Rpc方式调用API。

调用示例

调用示例接口信息：产品（ECS）、接口（DescribeInstances）、版本（2014-05-26）。

示例代码如下：

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkecs.request.v20140526 import DescribeInstancesRequest
import os

# 创建AcsClient实例
client = AcsClient(
    "<your-access-key-id>",
    "<your-access-key-secret>",
    "<your-region-id>",

    timeout=1000, #设置超时时间
    verify=False, # 忽略证书 或者 verify='<证书位置>'
);
# 和https一起使用
# client.set_verify(False)
# os.environ["HTTPS_PROXY"] = "http://127.0.0.1:8080"

# os.environ["HTTP_PROXY"] = "http://127.0.0.1:8080"
def hello_ecs_instance():
    # 创建request, 并设置参数
    request = DescribeInstancesRequest.DescribeInstancesRequest()

    params = {"PageNumber": "1", "Product": "Ecs", "Version": "2014-05-26", "Action": "DescribeInstances"};
    request.set_body_params(params)
    # No network protocol 格式为 "asapi域名/asapi/v3"
    request.set_endpoint('<ASAPI的Endpoint>')
    # 和endpoint一起使用, endpoint加了network protocol就不需要加下面的参数
    request.set_protocol_type("https")

    request.set_PageSize(10)
    # 发起API请求并显示返回值
    response = client.do_action_with_exception(request)
    print(response)

if __name__ == '__main__':
    print("Hello Aliyun OpenApi!")
    hello_ecs_instance()
```

设置代理

通过环境变量设置HTTP_PROXY、HTTPS_PROXY。

示例代码如下：

```
# https的代理
os.environ["HTTPS_PROXY"] = "http://x.x.x.x:xxxx"

# http的代理
os.environ["HTTP_PROXY"] = "https://x.x.x.x:xxxx"
```

1.5. HTTP调用方式

1.5.1. 概述

本文介绍HTTP调用方式，适用基于API URL发起HTTP GET请求的用户。

如果您使用的是SDK可以跳过此环节。

发起API请求的URL由不同参数拼凑而成，有固定的请求结构。URL中包含公共参数、您的签名和某个API的具体参数。每篇API文档均给出了URL请求示例供您参考，但是为了方便显示，我们并没有编码这些URL示例，您需要在发起请求前自行编码。我们根据您的签名验证了请求后，会返回结果给您。接口调用成功会显示返回参数，调用失败则显示相应报错，您可以根据错误信息分析排查。

1.5.2. 请求结构

专有云ASAPI支持基于URL发起HTTP GET请求。请求参数需要包含在URL中。本文列举了GET请求中的结构解释，并提供了ASAPI的服务接入地址（Endpoint）。

结构示例

以下为GetOrganizationList一条未编码的URL请求示例：

```
https://public.asapi.aliyuns.com/asapi/v3?Action=GetOrganizationList&Product=ascm&Version=2019-05-10&Id=<组织Id>&<公共请求参数>
```

- `https` 指定了请求通信协议。
- `public.asapi.aliyuns.com/asapi/v3` 指定了ASAPI的服务接入地址（Endpoint）。
- `Action=GetOrganizationList Product=ascm Version=2019-05-10` 指定了要调用的API信息。
- `Id` 是组织的ID。
- `<公共请求参数>` 是系统规定的公共参数。

通信协议

ASAPI只支持HTTPS协议请求通信。

接入地址

ASAPI服务接入地址，请参照[生成ASAPI的Endpoint](#)。

请求参数

您需要通过 `Action/Product/Version` 参数指定目标操作，例如

`Action=GetOrganizationList Product=ascm Version=2019-05-10`。还需要指定接口的其他参数以及公共请求参数，请参见[公共参数](#)。

1.5.3. 签名机制

对于每一次HTTP，我们会根据访问中的签名信息验证访问请求者身份。具体由使用AccessKey ID和AccessKey Secret对称加密验证实现。

说明

- Apsara Uni-manager运营控制台对每个用户和组织都生成了对应的AccessKey ID和AccessKey Secret。
- AccessKey ID是访问者身份，AccessKey Secret是加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密谨防泄露。
- ASAPI目前提供Node.js、Python和Java三种语言的SDK，可以免去您签名的烦恼。更多详情，请参照[获取ASAPI的core包](#)。

步骤一：构造规范化请求字符串

1. 排序参数。排序规则以首字母顺序排序，排序参数包括公共请求参数和接口自定义参数，不包括公共请求参数中的 `Signature` 参数。

说明

当使用GET方法提交请求时，这些参数就是请求URL中的参数部分，即URL中问号（?）之后由与号（&）连接的部分。

2. 编码参数。使用UTF-8字符集按照RFC3986规则编码请求参数和参数取值，编码规则如下：

- 字符英文大写或小写字母（A~Z、a~z）、阿拉伯数字（0~9）以及字符短划线（-）、下划线（_）、英文句号（.）、不编码。
- 其它字符编码成 `%XY` 的格式，其中 `XY` 是字符对应ASCII码的16进制。示例：半角双引号（"）对应 `%22`。
- 扩展的UTF-8字符，编码成 `%XY%ZA...` 的格式。
- 空格（ ）编码成 `%20`，而不是加号（+）。

该编码方式与 `application/x-www-form-urlencoded` MIME格式编码算法相似，但又有所不同。

如果您使用的是Java标准库中的 `java.net.URLEncoder`，可以先用标准库中 `percentEncode` 编码，随后将编码后的字符中加号（+）替换为 `%20`、星号（*）替换为 `%2A`、`%7E` 替换为波浪号（~），即可得到上述规则描述的编码字符串。

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedEncodingException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("'", "%2A")
        .replace("%7E", "~") : null;
}
```

3. 使用等号 (=) 连接编码后的请求参数和参数取值。
4. 使用与号 (&) 连接编码后的请求参数，注意参数排序与步骤1一致。

现在，您得到了规范化请求字符串 (CanonicalizedQueryString)，其结构遵循[请求结构](#)。

步骤二：构造签名字符串

1. 构造待签名字符串 `StringToSign`。您可以同样使用 `percentEncode` 处理上一步构造的规范化请求字符串，规则如下：

```
StringToSign=
HTTPMethod+"&"+//HTTPMethod: 发送请求的 HTTP 方法，例如 GET。
percentEncode("/")+"&"+//percentEncode("/")：字符 (/) UTF-8 编码得到的值，即 %2F。
percentEncode(CanonicalizedQueryString)//您的规范化请求字符串。
```

2. 按照[RFC2104](#)的定义，计算待签名字符串 `StringToSign` 的HMAC-SHA1或者HMAC-256值。示例使用的是Java Base64编码方法。

```
String signature = Base64.getEncoder().encodeToString(signData);
```

说明

计算签名时，RFC2104规定的Key值是您的 `AccessKeySecret` 再加上与号 (&)，其ASCII值为38。更多详情请参见[获取AccessKey](#)。

3. 添加根据[RFC3986](#)规则编码后的参数 `Signature` 到规范化请求字符串URL中。

示例一：参数拼接法

以调用 `GetOrganizationList` 查询地域为例。假设您获得了 `AccessKeyId=testid` 以及 `AccessKeySecret=testsecret`，签名流程如下所示：

1. 构造规范化请求字符串。

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

2. 构造待签名字符串 `StringToSign`。

```
GET&%2F&AccessKeyId%3Dtestid%26Action%3DGetOrganizationList%26Format%3DJSON%26Id%3D1%26Product%3Dascm%26RegionId%3Dcn-region-test%26SignatureMethod%3DHMAC-SHA1%26SignatureNonce%3D43967c6e-e3ad-44fa-b06b-fcf3db5f39d2%26SignatureVersion%3D1.0%26Timestamp%3D2020-09-15T08%253A50%253A40Z%26Version%3D2019-05-10
```

3. 计算签名值。

计算签名值。因为 `AccessKeySecret=testsecret`，用于计算的Key为 `testsecret&`，计算得到的签名值为 `OLeaidS1JvxuMvnyHOwuJ+uX5qY=`。示例使用的是JavaBase64编码方法。

```
String signature = Base64.getEncoder().encodeToString(signData);
```

 添加RFC3986规则编码后的

```
Signature=hvqv9Ll5RE6kQ4YsQXFaf9VuRIY=
```

 到步骤1的URL中。

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

通过以上URL，您可以使用浏览器、curl或者wget等工具发起HTTP请求调用 `DescribeRegions`，查看阿里云的地域列表。

示例二：编程语言法

依然以调用 `GetOrganizationList` 查询地域为例。假设您获得了 `AccessKeyId=testid` 以及 `AccessKeySecret=testsecret`，并且假定所有请求参数放在一个Java `Map<String, String>` 对象里。

1. 预定义编码方法。

```
private static final String ENCODING = "UTF-8";
private static String percentEncode(String value) throws UnsupportedOperationException {
    return value != null ? URLEncoder.encode(value, ENCODING).replace("+", "%20").replace("*", "%2A")
        .replace("%7E", "~") : null;
}
```

2. 预定义编码时间格式 `Timestamp`。参数 `Timestamp` 必须符合ISO8601规范，并需要使用UTC时间，时区为+0。

```
private static final String ISO8601_DATE_FORMAT = "yyyy-MM-dd'T'HH:mm:ss'Z'";
private static String formatIso8601Date(Date date) {
    SimpleDateFormat df = new SimpleDateFormat(ISO8601_DATE_FORMAT);
    df.setTimeZone(new SimpleTimeZone(0, "GMT"));
    return df.format(date);
}
```

3. 构造请求字符串。

```
final String HTTP_METHOD = "GET";
Map parameters = new HashMap();
// 输入请求参数
parameters.put("Product", "ascm");
parameters.put("Action", "GetOrganizationList");
parameters.put("Version", "2019-05-10");
parameters.put("AccessKeyId", "testid");
parameters.put("RegionId", "cn-region-test");
parameters.put("Timestamp", formatIso8601Date(new Date()));
parameters.put("SignatureMethod", "HMAC-SHA1");
parameters.put("SignatureVersion", "1.0");
parameters.put("SignatureNonce", UUID.randomUUID().toString());
parameters.put("Format", "JSON");
parameters.put("Id", "1");
// 排序请求参数
String[] sortedKeys = parameters.keySet().toArray(new String[{}]);
Arrays.sort(sortedKeys);
final String SEPARATOR = "&";
// 构造 stringToSign 字符串
StringBuilder stringToSign = new StringBuilder();
stringToSign.append(HTTP_METHOD).append(SEPARATOR);
stringToSign.append(percentEncode("/")).append(SEPARATOR);
StringBuilder canonicalizedQueryString = new StringBuilder();
for(String key : sortedKeys) {
    // 这里注意编码 key 和 value
    canonicalizedQueryString.append("&")
        .append(percentEncode(key)).append("=")
        .append(percentEncode(parameters.get(key)));
}
// 这里注意编码 canonicalizedQueryString
stringToSign.append(percentEncode(
    canonicalizedQueryString.toString().substring(1)));
```

4. 签名。因为 `AccessKeySecret=testsecret`，所以用于计算HMAC的Key为 `testsecret&`，计算得到的签名值为 `OLeaidS1JvxuMvnyHOwuJ%2BuX5qY%3D`。

```
// 以下是一段计算签名的示例代码
final String ALGORITHM = "HmacSHA1";
final String ENCODING = "UTF-8";
key = "testsecret&";
Mac mac = Mac.getInstance(ALGORITHM);
mac.init(new SecretKeySpec(key.getBytes(ENCODING), ALGORITHM));
byte[] signData = mac.doFinal(stringToSign.getBytes(ENCODING));
String signature = Base64.getEncoder().encodeToString(signData);
```

增加签名参数后，按照RFC3986规则编码后的URL如下所示：

```
https://public.asapi.aliyuns.com/asapi/v3/?AccessKeyId=testid&Action=GetOrganizationList&Format=JSON&Id=1&Product=ascm&RegionId=cn-region-test&SignatureMethod=HMAC-SHA1&SignatureNonce=43967c6e-e3ad-44fa-b06b-fcf3db5f39d2&SignatureVersion=1.0&Timestamp=2020-09-15T08%3A50%3A40Z&Version=2019-05-10
```

5. 使用浏览器、curl或者wget等工具发送HTTP请求。

1.5.4. 返回结果

返回结果目前只支持JSON结果的输出。当返回码大于等于200且小于300时为正常返回，否则为异常返回。

正常返回示例

接口调用成功后一般会返回请求ID和具体业务返回值，具体可参见云产品开发手册说明。

```
{
  "eagleEyeTracelId": "xxx-xxx-xxx-xxx", //鹰眼
  "asapiSuccess": true,
  "RequestId": "xxxxxxxxxx",
  "Regions": { // DescribeRegions 示例
    "Region": [
      {
        "RegionId": "",
        "RegionEndpoint": "",
        "LocalName": ""
      }
    ]
  }
}
```

异常返回示例

接口调用出错后，会返回如下结构，具体可参照云产品开发手册说明。

```
{
  "code": "asapi.server.api.notfound", //错误代码
  "eagleEyeTracelId": "xxx-xxx-xxx-xxx", //鹰眼
  "asapiSuccess": false, //本次请求是否成功
  "hostId": "xxxx", //主机ID
  "message": "Failed to ....", //消息
  "asapiErrorMessage": "Failed to ....", //ASAPI错误信息
  "asapiErrorCode": "asapi.server.api.notfound" //ASAPI错误码
}
```


2.API参考

2.1. 产品更新说明

本文介绍消息队列RocketMQ版（简称RocketMQ）从V3.8.0到V3.8.1的产品更新说明，帮助您更快地上手更新后的版本。

资源隔离优化（实例化支持）

RocketMQ开放实例化功能，解决多租之间的隔离问题，每个用户可以购买多个实例，每个实例之间逻辑隔离。

为兼容老用户的既有资源，RocketMQ实例的命名空间分为两种：

- 默认实例（兼容老用户的既有资源）
 - 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一。
 - 对RocketMQ老用户来说，既有资源将自动生成一个默认的无独立命名空间实例；无既有资源，则可选择创建不带命名空间的实例，最多创建1个。
 - 接入点配置（可在控制台实例详情页面中获取）。

```
// 推荐配置
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
// 兼容配置（不推荐继续使用，建议逐渐升级为推荐配置）
properties.put(PropertyKeyConst.ONSAddr, "xxx");
```

- 新建实例
 - 有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名。
 - 接入点配置（可在控制台实例详情页面中获取）。

```
// 推荐配置
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
```

- ons-client客户端必须升级到最新版本：
 - JAVA：1.8.7.1.Final
 - C/C++：v2.0.0
 - .NET：1.1.3

资源申请流程优化

原RocketMQ的资源包括3个部分：消息主题（Topic）、生产者（Producer ID）、消费者（Consumer ID），三者之间都是多对多的对应关系，三者之间的关系复杂而难以理解；对于中、大型企业客户来说，每创建一个Topic，都要关联或者更新Producer ID和Consumer ID，流程过于复杂。

因此，为进一步优化用户体验，降低新用户的接入门槛，对资源申请的接入流程进行简化。

资源申请方面，主要包括以下部分：

- Topic管理（保持不变）
 - Topic资源申请，一级消息类型，通过Topic对消息进行分类。
- Group管理
 - 取消原Producer ID的申请，与原Consumer ID统一为Group ID，即对应控制台上，取消原生产者管理，与原消费者管理统一为Group管理。

- 取消原申请Producer ID或Consumer ID时需要指定Topic的关联约束，只需申请Group ID即可在代码中直接配置与Topic的关联使用。
- 兼容性相关问题
 - 原申请的Producer ID列表将不再显示，不影响当前业务的正常使用。
 - 原申请的以“CID-”或者“CID_”打头的Consumer ID仍可以正常使用，代码中既可在PropertyKeyConst.ConsumerId中设置，亦可在PropertyKeyConst.GROUP_ID中设置。
- 示例代码

② 说明

- 建议您将ons-client客户端升级为最新版本：
 - JAVA: **1.8.7.1.Final**
 - C/C++: **v2.0.0**
 - .NET: **1.1.3**
- 原来已经在使用的Producer ID或者Consumer ID保持兼容，不影响当前在线业务的正常使用，但建议逐步替换为新的推荐配置方式。

- 推荐方式：将Producer ID、Consumer ID的概念融合成Group ID。

```
// 设置PropertyKeyConst.GROUP_ID, 原PropertyKeyConst.ProducerId和PropertyKeyConst.ConsumerId会被@Deprecated
properties.put(PropertyKeyConst.GROUP_ID, "原CID-XXX或GID-XXX");
```

- 兼容方式：Producer ID用来标识生产者，Consumer ID用来标识消费者。

```
// 创建Producer时需要配置PropertyKeyConst.ProducerId。
properties.put(PropertyKeyConst.ProducerId, "原PID-XXX或GID-XXX");
// 创建Consumer时需要配置PropertyKeyConst.ConsumerId。
properties.put(PropertyKeyConst.ConsumerId, "原CID-XXX或GID-XXX");
```

2.2. RocketMQ Open API简介

2.2.1. 面向用户和场景

Open API是RocketMQ提供给用户的管控手段，用于实现一系列资源管理和运维功能。同生产用SDK不同，Open API主要用于管理RocketMQ相关的资源，例如Topic、Group ID等。

Open API的主要用户：

- 已经熟悉RocketMQ控制台的主要功能。
- 需要实现类似控制台的管控，存在自定义管控功能等需求。

2.2.2. 适用场景和局限性

1. Open API由于其实现原理，调用端必须具备访问RocketMQ控制台域名能力，否则无法调通。
2. Open API底层基于HTTP协议实现，性能和效率不高，仅适用于管控场景，请勿用于大批量高并发的生产场景。

2.2.3. Open API基本原理

Open API底层通过HTTP接口提供服务，用户调用HTTP客户端，发出HTTP请求到RocketMQ的后端服务去执行。

用户将参数封装到每个请求中，每个请求即对应一个接口，执行的结果放在response中，请求和响应的字符编码都是采用UTF-8编码。

其中，HTTP请求的服务端地址即为控制台的域名地址。

2.3. RocketMQ Open API接入指南

2.3.1. 公共参数设置

客户端基于HTTPS方式调用RocketMQ Open API，所有API请求都需要传入一部分公共参数，供服务端进行权限校验。

公共参数列表如下：

参数名称	类型	是否必须	说明
_accesskey	String	是	用户账号AK
__preventCache	Long	是	用于CSRF校验，设置为系统当前时间即可，单位毫秒（ms）
_regionId	String	是（获取regionId列表接口除外）	当前Region
_signature	String	是	请求的签名


_regionId参数需要填写当前请求操作的Region，专有域和专有云情况下，默认值为mq-private。

_signature参数是对HTTPS请求中其他参数字段做排序后得到待签名字符串，然后使用账号的SecretKey进行签名计算得到。

签名计算示例

例如原始请求的HTTP方法是：

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

 **说明** 在单region部署的情况下或接入多region部署中的中心region，使用\${global:internet-domain}作为访问域名；接入多region部署下的普通region，使用\${global:region}.\${global:internet-domain}作为访问域名

参数列表有（参数包含公共参数和每个接口自定义的参数）：

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

那么原始请求URL是：

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

将Key_2, Key_1, Key_3这3对参数的key按字典序排序, 然后将对应的value按字符串连接。Key的排序结果为: Key_1、Key_2、Key_3, 排序后其对应值的连接结果为Value1Value2Value3。此时待签名字符串就是Value1Value2Value3。

然后使用您的secretKey作为密钥, 使用 HmacSHA1方法对上面的待签名字符串做签名, 得到签名字符串。Hmac的算法实现, 各个语言都有现成的函数库, 请自行搜索。

Linux命令行的方式可以参考:

```
echo -n 'Value1Value2Value3' | openssl dgst -hmac '用户的SecretKey' -sha1 -binary | base64
```

假设最终得到的结果为: eqweq+adwe23****=

最后, 整个HTTPS请求就是(注意通过curl的方式需要将“+”替换成“%2B”, “=”替换成“%3D”, “/”替换成“%2F”, 否则自动转义后会报鉴权失败):

```
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
```

2.3.2. 公共返回值解析

客户端发起HTTP请求到服务端, 服务端处理完成后同样以HTTP响应形式返回结果, 正常情况下返回结果HTTP状态码都是200, 返回的内容放在Body中, 为JSON字符串, 需要客户端自己做JSON解析, 判断是否调用成功。

JSON格式如下:

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

返回值说明:

- code: 状态码, 200代表成功, 其他都是处理失败。
- currentPage: 当前请求如果是分页查询的请求, 此变量代表结果所处的当前页。
- dynamicMessage: 错误异常信息字符串。
- message: 处理结果信息。

- pageSize: 当前请求如果是分页查询的请求，此变量代表结果所处的当前页大小。
- success: 是否成功。
- data: 业务信息返回，String类型，也是JSON，如果当前请求的返回结果中带有业务数据，则此处需要反解JSON内容。
- RequestId: 每个请求的唯一ID，String类型。

2.4. RocketMQ Open API公共术语

2.4.1. 账号安全类

2.4.1.1. AccessKey

访问阿里云产品使用的账号的AccessKeyId，在控制台获得。调用时传入给RocketMQ后台用于身份验证，调用RocketMQ Open API只能操作对应账号下的资源。

2.4.1.2. SecretKey

访问阿里云产品使用的账号的AccessKeySecret，在控制台获得，调用时传入，用于计算访问请求的签名。RocketMQ后端根据对应账号比对签名实现鉴权。

2.4.2. RocketMQ资源管控类

2.4.2.1. Topic

RocketMQ产品中消息按照Topic进行分类存储，因此Topic是RocketMQ中最重要的资源之一。RocketMQ中Topic全局唯一。

2.4.2.2. Group ID (GID)

一类Producer或Consumer，这类Producer或Consumer通常生产或消费同一类消息，且消息发布或订阅的逻辑一致。

- 对于子账号创建的消费者Group ID或存量CID，该子账号及其主账号都有使用权限。
- 对于主账号创建的消费者Group ID或存量CID，仅该主账号有使用权限，其子账号无使用权限。
- 存量Producer ID和Consumer ID与Group ID的使用变更，请参见[产品更新说明](#)。

存量Producer ID和Consumer ID与Group ID的使用变更，请参见[产品更新说明](#)。

2.5. RocketMQ Open API错误码列表

RocketMQ Open API调用成功时返回给客户端的ResponseCode=200，当出现调用失败时，会返回对应的异常错误码以及描述信息。您可以根据以下异常信息对照表查找对应的解决方法。

错误码	现象&原因	解决方法
ONS_SYSTEM_ERROR	RocketMQ后端异常	通过阿里云工单联系RocketMQ技术人员

错误码	现象&原因	解决方法
ONS_SERVICE_UNSUPPORTED	当前调用在对应的区域不支持	请找RocketMQ技术支持确认该接口的开通情况
ONS_INVOKE_ERROR	Open API接口调用失败	联系RocketMQ技术人员处理
BIZ_FIELD_CHECK_INVALID	参数检验失败	请参考API手册检查各个参数传入是否合法
BIZ_TOPIC_NOT_FOUND	Topic没有找到	请检查Topic输入是否合法，或者是否创建过
BIZ_SUBSCRIPTION_NOT_FOUND	目标订阅关系GID找不到	请检查GID是否创建过，或者查询条件错误
BIZ_PUBLISHER_EXISTED	指定GID已经存在	更换GID的名称重试请求
BIZ_SUBSCRIPTION_EXISTED	指定GID已经存在	更换GID的名称重试请求
BIZ_CONSUMER_NOT_ONLINE	指定GID的客户端不在线	请确保消费端在线，然后重试请求
BIZ_NO_MESSAGE	当前查询条件没有匹配消息	请检查查询条件，并确认查询范围内是否发过消息
BIZ_REGION_NOT_FOUND	请求的region找不到	确认请求的Region参数是否合法
BIZ_TOPIC_EXISTED	指定Topic已经存在	更改Topic名称重试请求
BIZ_PRODUCER_ID_BELONG_TO_OTHER_USER	当前GID已经被其他用户占用	更换GID重试请求
BIZ_CONSUMER_ID_BELONG_TO_OTHER_USER	当前GID已经被其他用户占用	更换GID重试请求
BIZ_PUBLISH_INFO_NOT_FOUND	请求的GID没有找到	确认GID是否存在，或者请求条件错误
EMPOWER_EXIST_ERROR	当前授权关系已经存在	核实请求参数，重试或者先查询
EMPOWER_OWNER_CHECK_ERROR	当前用户不是授权Topic的Owner	确认资源所属关系

2.6. API详细说明

2.6.1. Region管理

2.6.1.1. 获取regionId列表 (/json/region/list)

本节介绍获取regionId列表的接口，该接口用于获取regionId的列表，获取的regionId将作为后续接口的公共参数_regionId。

使用场景

获取regionId的列表，获取的regionId将作为后续接口的公共参数_regionId。

请求参数列表

请参考[公共参数设置](#)。

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	List(RegionDo)	查询结果

RegionDo数据结构列表

名称	类型	描述
id	Long	数据库中编号
regionId	String	地域的ID，例如“cn-hangzhou”
regionName	String	地域的名称，例如“华东1（杭州）”

2.6.2. 集群管理

2.6.2.1. 获取集群信息（/json/cluster/list）

本文介绍获取集群信息的接口，您在创建实例时需要传入集群名称，以指定实例和集群间的所属关系。集群名称可通过本接口获取。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
_regionId	String	是	指定查询哪个地域下的集群信息

返回参数列表

请参考[公共返回值解析](#)。

名称	类型	描述
data	List<ClusterInfoDo>	查询结果

ClusterInfoDo数据结构列表

名称	类型	描述
regionId	String	查询到的集群所在的地域
cluster	String	查询到的集群
createTime	String	查询到的集群的创建时间
modifyTime	String	查询到的集群的最近一次修改时间

2.6.3. 实例管理

2.6.3.1. 实例网络类型切换

(/json/vpcTunnel/tunnelSwitch)

集群VPC网络类型，为mq-broker、mq-nameserver两个组件申请指定类型的VIP。

使用场景

切换实例的访问VPC网络类型，支持Single Tunnel和Any Tunnel两种方式。

注意事项：

- 网络切换针对指定的集群进行，会影响该集群中正在收发消息的所有Topic，因此调用之前一定要确认影响范围。
- 切换集群VPC网络类型后，需要将连接该集群的所有客户端的配置修改为接口返回值：`properties.put(PropertyKeyConst.NAMESRV_ADDR,"接口返回值");`。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
tunnelType	String	是	VPC网络类型，anyTunnel或者singleTunnel
instanceId	String	是	需要切换的实例的ID
slbUrl	String	否	SLB的API

名称	类型	是否必须	描述
force	Boolean	否	值为True时，不管集群的Broker是否有消息流量，强制进行网络切换；值为False时，如果Broker的单机InTps + OutTps > 10，阻止本次切换。默认值为False。
vSwitchId	String	否	tunnelType为singleTunnel时有效，表明将singleTunnel的VIP建在用户的某个VPC实例上。如果用户只有一个VPC或者tunnelType为anyTunnel时不需要指定该参数。

返回参数列表

以JSON格式返回NameServer的VIP地址，需要将连接该集群的所有客户端配置为：
`properties.put(PropertyKeyConst.NAMESRV_ADDR,"接口返回值");`，并重启应用使配置生效。

2.6.3.2. 获取实例基本信息 (/json/instance/baseInfo)

用于查询实例基本信息，包括收发消息的接入点。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	实例ID

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
InstanceBaseInfo	InstanceBaseInfo	实例信息对象

InstanceBaseInfo数据结构列表

名称	类型	描述
instanceId	String	实例ID
instanceName	String	实例名称

名称	类型	描述
independentNaming	Boolean	<p>是否有命名空间。取值说明如下：</p> <ul style="list-style-type: none"> • true: 拥有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名 • false: 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一 <p>命名空间可在RocketMQ控制台实例详情页面查看。命名空间的详情请参见产品更新说明。</p>
Endpoints	Endpoints	各个协议的接入点
maxSendTps	Long	单个实例的消息发送TPS上限
maxReceiveTps	Long	单个实例的消息消费TPS上限
topicCapacity	Integer	单个实例允许创建的Topic数量上限

Endpoints数据结构列表

名称	类型	是否为空	描述
tcpEndpoint	String	否	RocketMQ的TCP协议接入点

2.6.3.3. 创建实例（/json/instance/create）

创建实例。

说明

- 默认仅限主账号使用本接口，子账号只有在被授予了相关权限后方可使用。
- 每个地域（Region）最大的实例数量为500。
- 实例名称在该实例所部署的地域下要唯一，不可重名。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
cluster	String	是	指定实例创建于哪个集群
instanceName	String	是	实例名称，长度为3-64，中英文都可以

名称	类型	是否必需	描述
independentNaming	Boolean	是	<p>是否有命名空间。取值说明如下：</p> <ul style="list-style-type: none"> • true: 拥有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名 • false: 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一 <p>命名空间可在RocketMQ控制台实例详情页面查看。命名空间的详情请参见产品更新说明。</p>
remark	String	否	备注，可以不填
maxSendTps	Long	是	单个实例的消息发送TPS上限
maxReceiveTps	Long	是	单个实例的消息消费TPS上限
topicCapacity	Integer	否	单个实例允许创建的Topic数量上限

返回参数列表

请参考[公共返回值解析](#)。

2.6.3.4. 删除实例 (/json/instance/delete)

删除实例。

② 说明

- 默认仅限主账号使用本接口，子账号只有在被授予了相关权限后方可使用。
- 实例下Topic和GroupId参数为空时才可删除实例。

实例下Topic和GroupId参数为空时才可删除实例。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	实例ID

返回参数列表

请参考[公共返回值解析](#)。

2.6.3.5. 获取实例列表 (/json/instance/list)

获取当前环境所有实例的列表。

使用场景

查询所有的实例信息，供进行实例管理。

请求参数列表

请参考[公共参数设置](#)。

返回参数列表

请参考[公共返回值解析](#)。

名称	类型	描述
data	List(InstanceVO)	返回所有已发布的实例信息列表

InstanceVO数据结构列表

名称	类型	描述
instanceId	String	实例ID
instanceName	String	实例名称
namespaceRulesType	Boolean	<p>是否有命名空间。取值说明如下：</p> <ul style="list-style-type: none"> • true: 拥有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名 • false: 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一 <p>命名空间可在RocketMQ控制台实例详情页面查看。命名空间的详情请参见产品更新说明。</p>

2.6.3.6. 更新实例名称和描述 (json/instance/update)

更新实例名称和描述。

说明

- 默认仅限主账号使用本接口，子账号只有在被授予了相关权限后方可使用。
- 每个地域（Region）最大的实例数量为500。
- 实例名称在该实例所部署的地域下要唯一，不可重名。

请求参数列表

其他请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	实例ID
instanceName	String	否	实例名称，长度3-64，中英文都可以；传空值则不修改 说明 同一地域下的实例名称必须唯一。
remark	String	否	备注，传空值则不修改

返回参数

请参考[公共返回值解析](#)。

2.6.4. Topic管理**2.6.4.1. 创建Topic (/json/topic/create)**

创建Topic接口，该接口用于创建指定Topic到目标地域上。

使用场景

本接口一般是在发布新应用、业务逻辑实现等需要新Topic的情况下使用。

默认仅限主账号使用本接口，子账号只有在被授予了相关权限后方可使用。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
----	----	------	----

名称	类型	是否必须	描述
topic	String	是	<p>需要创建的Topic名称。</p> <p>说明</p> <ul style="list-style-type: none"> “CID”和“GID”是Group ID的保留字段，Topic命名不能以“CID”和“GID”开头。 如果创建Topic所在的实例有命名空间，则新建的Topic命名只需确保实例内唯一，跨实例之间可重名。 如果新建的Topic所在的实例没有命名空间，则新建的Topic命名需保证跨实例和跨地域全局唯一。
instanceId	String	是	<p>需创建的Topic所对应的实例ID。</p> <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 <p>实例命名空间详情请参见产品更新说明。</p>
orderType	Integer	是	<p>消息类型。</p> <p>说明</p> <ul style="list-style-type: none"> 0：普通消息 1：分区顺序消息 2：全局顺序消息 4：事务消息 5：定时/延时消息
remark	String	否	备注，可以不填

返回参数列表

请参考[公共返回值解析](#)。

使用示例

_signature计算方式（请参考[公共参数设置](#)）：

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

调用方式：

```
curl http://mq.console.example.com//json/topic/create?_accesskey=ACCESSKEY_XXX&__preventCache=1498029701000&_platform=onsConsole&_regionId=mq-private&topic=TOPIC_XXX&_signature=8DqV%2BrOBzOd0M2SNj3Acuqn****%3D
```

 **说明** example.com是项目自定义的访问域名，请联系管理员获取该域名。

返回结果：

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

2.6.4.2. 删除Topic (/json/topic/delete)

此接口用于删除账号下的指定Topic。

使用场景

本接口一般是在应用下线等资源回收场景下使用。Topic删除时RocketMQ后台需要做资源回收，速度较慢，因此不建议删除后立即又重新创建。如果删除失败，请根据错误码进行处理。

 **注意** 删除Topic将导致该Topic下所有的发布订阅关系同时被清理，请慎重调用。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
topic	String	是	指定删除的Topic
instanceId	String	是	需删除的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。

返回参数列表

请参考[公共返回值解析](#)。

2.6.4.3. 查询Topic (/json/topic/get)

查询某个Topic具体信息的接口。

使用场景

用来查询Topic的基本信息，如owner、状态等。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
topic	String	是	指定查询的Topic
instanceId	String	是	需删除的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。

返回参数列表

请参考[公共返回值解析](#)。

2.6.4.4. 查询Topic列表 (/json/topic/listInPage)

此接口用于按页查询该用户的Topic列表。

使用场景

按照分页的方式来查询用户的Topic列表，一般先通过listNames接口获取所有Topic的名字，然后调用该接口获取Topic的详细信息。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
topic	String	否	指定查询的Topic
currentPage	Integer	否（默认1）	当前查询的页数
pageSize	Integer	否（默认20）	每页的Topic数量
instanceId	String	是	查询指定实例的Topic列表

返回参数列表

以JSON格式返回对应页数与条数的Topic的列表。

2.6.4.5. 配置Topic读写权限 (json/topic/update)

配置Topic的读写模式。

使用场景

默认仅限主账号使用本接口，子账号只有在被授予了相关权限后方可使用。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需配置的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需配置的Topic名称，在一个用户下不可重复，即使地域（Region）不同
perm	Integer	是	设置该Topic的读写模式。取值说明如下： <ul style="list-style-type: none"> 6：同时支持读写 4：禁写 2：禁读

返回参数列表

请参考[公共返回值解析](#)。

2.6.4.6. 查询Topic名字（/json/topic/listNames）

此接口用于查询某用户的所有Topic名字，一般与下文的listInPage接口配合使用查询所有Topic的详细信息。

使用场景

查询账号下所有Topic的名字。

请求参数列表

请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	需查询名字的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。

返回参数列表

以JSON的格式返回该用户创建的所有Topic的名字。

2.6.4.7. 查询Topic的订阅关系（json/topic/subDetail）

本接口只能用于查看有哪些在线的订阅组订阅了指定的Topic。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需查询的Topic

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	返回Data对象

Data数据结构列表

名称	类型	描述
topic	String	Topic
SubscriptionDataList	List<SubscriptionDataListItem>	所有在线订阅组

SubscriptionDataListItem数据结构列表

名称	类型	描述
groupId	String	订阅该Topic的Group ID
subString	String	订阅表达式
messageModel	String	消费模式。取值说明如下： <ul style="list-style-type: none"> CLUSTERING：集群订阅 BROADCASTING：广播订阅

2.6.5. Group管理

2.6.5.1. 创建Group (json/group/create)

创建客户端Group ID。

使用说明

在使用本接口时，需注意：

- 对于子账号创建的消费者Group ID或存量CID，该子账号及其主账号都有使用权限。
- 对于主账号创建的消费者Group ID或存量CID，仅该主账号有使用权限，其子账号无使用权限。
- 存量Producer ID和Consumer ID与Group ID的使用变更，请参见[产品更新说明](#)。

在使用本接口时，需注意存量Producer ID和Consumer ID与Group ID的使用变更，请参见[产品更新说明](#)。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceld	String	是	需创建的Group ID所对应的实例ID
groupid	String	是	创建的消息消费集群的Group ID。取值说明如下： <ul style="list-style-type: none"> • 以“GID_”或者“GID-”开头，只能包含字母、数字、短划线(-)和下划线(_)。 • 长度限制在5-64字符之间。 <div style="border: 1px solid #add8e6; padding: 5px; margin-top: 10px;"> <p> 说明 Group ID一旦创建，则无法修改。</p> </div>
remark	String	否	备注

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
remark	String	备注

2.6.5.2. 删除Group (json/group/delete)

删除之前使用OnsGroupCreate接口创建的Group。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需删除的Group ID对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupId	String	是	需要删除的消息消费集群的Group ID

返回参数列表

请参考[公共返回值解析](#)。

2.6.5.3. 获取Group信息列表 (json/group/list)

展示您的Group ID资源的列表，并不用于具体的信息查询。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需获取资源列表的Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupId	String	否	查询指定Group ID时设置，否则查询所有Group ID

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	List(SubscribeInfoDo)	查询到的订阅关系集合

SubscribeInfoDo数据结构列表

名称	类型	描述
owner	String	该发布信息的拥有者
instanceId	String	实例ID
groupId	String	消费集群ID
independentNaming	Boolean	<p>是否有命名空间。取值说明如下：</p> <ul style="list-style-type: none"> • true: 拥有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名 • false: 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一 <p>命名空间可在RocketMQ控制台实例详情页面查看。命名空间的详情请参见产品更新说明。</p>
remark	String	备注

2.6.5.4. 查询Group订阅关系 (json/group/subDetail)

本接口只能用于查看指定的、在线的Group订阅了哪些Topic，如果Group不在线则查不到数据。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	<p>需查询的Group ID所属的实例ID。</p> <ul style="list-style-type: none"> • 如果您的实例无命名空间，则无需填写该参数。 • 如果您的实例有命名空间，则该参数为必填。 <p>实例命名空间详情请参见产品更新说明。</p>
groupId	String	是	需查询的Group ID

返回参数列表

其他请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	返回Data对象

Data数据结构列表

名称	类型	描述
groupId	String	需查询的Group ID
online	Boolean	是否在线
messageModel	String	消费模式。取值说明如下： <ul style="list-style-type: none"> • CLUSTERING：集群订阅 • BROADCASTING：广播订阅
SubscriptionDataList	List<SubscriptionDataListItem>	如果Group不在线则查不到数据

SubscriptionDataListItem数据结构列表

名称	类型	描述
topic	String	Topic
subString	String	订阅表达式

2.6.5.5. 配置订阅权限 (json/group/consumerUpdate)

配置指定 Group ID 对应的消费集群的消息读取权限。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需配置的Group ID所对应的实例ID。 <ul style="list-style-type: none"> • 如果您的实例无命名空间，则无需填写该参数。 • 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。

名称	类型	是否必需	描述
groupid	String	是	需配置的Group ID名称
readEnable	Boolean	是	设置读消息开关

返回参数列表

其他请参考[公共返回值解析](#)。

2.6.6. 消费管理

2.6.6.1. 查询消费堆积（/json/consumer/accumulate）

此接口用于查询指定订阅组的消费堆积情况，可以获取到订阅组当前未消费的消息数以及延迟时间。

使用场景

消费堆积查询一般在生产环境中需要关注订阅组消费进度时使用，用于判断消息消费情况和延迟情况。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	需查询的Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupid	String	是	需查询的消费端Group ID
detail	Boolean	否	是否返回详细的堆积信息，包含单个队列中的消息堆积详情

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	指定Consumer的消费堆积情况

Data数据结构列表

名称	类型	描述
----	----	----

名称	类型	描述
onLine	Boolean	该Group ID下是否有消费者实例在线；只要有一个消费者实例在线，该Group ID状态即为在线。取值说明如下： <ul style="list-style-type: none"> • true：状态为在线 • false：状态为不在线
totalDiff	Long	当前Group ID订阅的所有Topic的消息总堆积数量
consumeTps	Float	该Group ID下消费者实例群组接收消息的总TPS
lastTimestamp	Long	该Group ID下消费者实例群组中最近消费的那条信息的生产时间
delayTime	Long	该Group ID下消费者实例群组中消息消费的最大延迟时间
DetailInTopicList	List(OnsConsumerAccumulateResponse.Data.DetailInTopicDo)	各个Topic具体情况

DetailInTopicList数据结构列表

名称	类型	描述
topic	String	Topic名称
totalDiff	Long	该Topic的消费堆积数量
lastTimestamp	Long	该Topic中最近一条被消费的信息的生产时间
delayTime	Long	该Topic所发消息的最大消费延迟时间

2.6.6.2. 查询消费状态 (/json/consumer/status)

查询指定Group ID的详细状态数据，包含订阅关系检查、消费TPS统计、负载均衡状态、消费端连接等。

使用场景

注意

- 本接口一般用于在粗略判断消费堆积以及客户端在线情况后，需要排查消费异常的原因时调用，可以判断出指定Group ID的订阅关系是否一致、负载均衡是否正常以及获取在线客户端的Jstack信息等。
- 本接口需要调用后端大量接口完成数据聚合，因此查询速度慢，不建议频繁调用。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	需查询的Group ID所对应的实例ID。 <ul style="list-style-type: none"> ● 如果您的实例无命名空间，则无需填写该参数。 ● 如果您的实例有命名空间，则该参数为必填。实例命名空间详情请参见产品更新说明。
groupId	String	是	需要查询的消费端Group ID
detail	Boolean	否	是否查询详细信息。取值说明如下： <ul style="list-style-type: none"> ● true: 要查询详细信息，即可获得返回参数ConsumerConnectionInfoList和DetailInTopicList信息。 ● false: 默认值，不查询详细信息，即返回参数ConsumerConnectionInfoList和DetailInTopicList为空。
needJstack	Boolean	否	是否打印Jstack信息。取值说明如下： <ul style="list-style-type: none"> ● true: 要打印Jstack信息，可在Jstack返回参数中获取堆栈信息。 ● false: 默认值，不打印Jstack信息，Jstack返回参数为空。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> 说明 如需打印Jstack信息，请确保您的入参Detail设置为true。</p> </div>

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	ConsumerStatusDo	查询结果

ConsumerStatusDo数据结构列表

名称	类型	描述
online	Boolean	是否在线
totalDiff	Long	集群总的消费堆积
consumeTps	Long	总消费TPS
lastTimestamp	Long	最后消费时间
delayTime	Long	延迟时间
consumeModel	Long	消费模型
subscriptionSame	Boolean	订阅关系是否一致
rebalanceOK	Boolean	客户端Rebalance是否正常。取值说明如下： <ul style="list-style-type: none"> • true: 正常 • false: 不正常
ConnectionSet	List(ConnectionDo)	该集群当前在线客户端信息
DetailInTopicList	List(DetailInTopicDo)	各个Topic的消费情况。如需获取详细信息，请确保将Detail入参设置为true。否则返回值为空。
ConsumerConnectionInfoList	List(ConsumerConnectionInfoDo)	该集群在线客户端详细信息，包含Jstack、消费RT时间等信息。如需获取详细信息，请确保将Detail入参设置为true。否则返回值为空。

ConnectionDo数据结构列表

名称	类型	描述
clientId	String	消费实例的ID
clientAddr	String	该消费实例的地址和端口
language	String	消费端语言
version	String	消费端版本
remoteIP	String	宿主机IP或公网IP

DetailInTopicDo数据结构列表

名称	类型	描述
topic	String	Topic名称

名称	类型	描述
totalDiff	Long	该Topic消费总堆积数
lastTimestamp	Long	最后消费时间
delayTime	Long	延迟时间

ConsumerConnectionInfoDo数据结构列表

名称	类型	描述
clientId	String	消费实例的ID
connection	String	连接信息
language	String	客户端语言
version	String	客户端版本号
consumeModel	String	消费类型。取值说明如下： <ul style="list-style-type: none"> • CLUSTERING：集群消费模式 • BROADCASTING：广播消费模式
consumeType	String	消费者消费消息的模式。取值说明如下： <ul style="list-style-type: none"> • PUSH：RocketMQ服务端主动向消费者推送消息。 • PULL：消费者主动向RocketMQ服务端拉取消息。
threadCount	Integer	消费线程数
startTimeStamp	Long	开始消费时间
lastTimeStamp	Long	最后消费时间
SubscriptionSet	List(SubscriptionData)	订阅关系集合
RunningDataList	List(ConsumerRunningDataDo)	实时状态统计
Jstack	List(ThreadTrackDo)	Jstack堆栈信息

SubscriptionData数据结构列表

名称	类型	描述
topic	String	订阅的Topic名称
subString	String	订阅该Topic的子类别Tag表达式

名称	类型	描述
subVersion	Long	订阅关系版本号，为自增Long型
tagsSet	List(String)	订阅的Tag集合

ConsumerRunningDataDo数据结构列表

名称	类型	描述
groupId	String	订阅方的Group ID
topic	String	订阅的Topic名称
rt	Float	消费RT时间，单位ms
okTps	Float	消费消息成功的TPS统计
failedTps	Float	消费消息失败的TPS统计
failedCountPerHour	Long	每小时内消费失败的消息数统计

ThreadTrackDo数据结构列表

名称	类型	描述
thread	String	线程名称
trackList	List(String)	Jstack堆栈信息字符串

2.6.6.3. 查询消费端连接信息

(/json/consumer/getConnection)

查询指定Group ID下当前客户端的连接情况。

使用场景

本接口一般用于判断指定Group ID的消费者是否在线，并获取详细的客户端连接的列表。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
instanceId	String	是	需查询的Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupId	String	是	需查询的Group ID

返回参数列表

名称	类型	描述
data	List(ConnectionDo)	指定Group ID的连接信息

ConnectionDo数据结构列表

成员	类型	描述
clientId	String	消费实例的ID
clientAddr	String	该消费实例的地址和端口
language	String	消费端语言
version	String	消费端版本
remoteIP	String	宿主机IP或公网IP

2.6.6.4. 查询消息可重置的时间范围

(/json/consumer/timespan)

查询当前Group ID订阅的Topic的最新消息时间戳以及消费的最新时间。

使用场景

使用本接口可以查询某个Topic当前服务器上存在的消息的最新时间和最早时间，以及查询消费端最近消费的时间。本接口一般用于展示消费进度的概况，和OnsConsumerAccumulate接口配合使用。

OnsConsumerAccumulate接口的详细信息，请参见[消费堆积查询 \(/json/consumer/accumulate\)](#)。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询的消费端Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupId	String	是	需查询的消费端Group ID
topic	String	是	需查询的消费端Group ID订阅的Topic

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	查询结果

Data数据结构列表

名称	类型	描述
topic	String	需查询的Topic名称
minTimeStamp	Long	整个Topic目前存储的最新的消息的时间
maxTimeStamp	Long	整个Topic目前存储的最早的消息的时间
consumeTimeStamp	Long	当前Group消费该Topic的最新的的时间

2.6.6.5. 重置消费位点 (/json/consumer/resetOffset)

重置指定的Group ID的消费位点到指定时间戳。

使用场景

本接口一般用于清理堆积消息，或者回溯消费。有以下两种清理方式：

- 清理所有消息
- 清理消费进度到指定的时间

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询的Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
groupId	String	是	需查询的消费端Group ID
topic	String	是	需重置位点的Topic 名称
type	Integer	是	消息清理类型。取值说明如下： <ul style="list-style-type: none"> 0：清除所有消息 1：清理到指定时间
resetTime	Long	false	重置位点到指定的时戳，仅仅在Type为1时生效

返回参数列表

请参考[公共返回值解析](#)。

2.6.7. 消息查询

2.6.7.1. 查询消息投递状态（/json/message/trace）

根据Message ID来判断目标消息是否曾被消费过。

使用场景

本接口的实现基于RocketMQ的内部位点机制，在大部分场景下判断结果可信，但如果应用方进行过重置位点已经清除消息等操作，则判断结果可能有误。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
instanceId	String	是	需查询的消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	消息的Topic
msgId	String	是	消息ID，即Message ID

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	List(MessageTrack)	查询结果

MessageTrack数据结构列表

名称	类型	描述
instanceId	String	实例ID
consumerGroup	String	消费集群名称
trackType	String	当前状态。取值如下： <ul style="list-style-type: none"> CONSUMED：已消费 CONSUMED_BUT_FILTERED：已被过滤 NOT_CONSUME_YET：暂未消费 NOT_ONLINE：客户端不在线 UNKNOWN：其他问题
exceptionDesc	String	异常描述

2.6.7.2. 根据Message ID查询消息

(/json/message/getById)

通过传入Message ID查询指定消息的信息以及判断该指定的消息是否曾被消费过。查询到的信息包括发送时间、存储服务器和消息的Key和Tag等属性。

使用场景

根据Message ID查询消息的方式属于精确查询，查询条件需要的Message ID从每次消息发送成功的SendResult中获取，因此业务方必须存储每次发送消息的结果。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
msgId	String	是	需查询的消息的ID，即Message ID
topic	String	是	需查询的消息的Topic

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	查询结果

Data数据结构列表

名称	类型	描述
topic	String	消息的Topic
propertyList	List(MessageProperty)	消息属性列表
body	String	消息体
storeSize	Integer	消息大小
bornTimestamp	Long	生成时间戳
bornHost	String	生成该消息的客户端实例
storeTimestamp	Long	被服务端存储的时间戳
storeHost	String	存储该消息的服务器实例

名称	类型	描述
msgId	String	消息ID, 即Message ID
bodyCRC	Integer	消息体CRC校验值
reconsumeTimes	Integer	消息重试消费的次数

2.6.7.3. 根据Message Key查询消息 (/json/message/getByKey)

通过传入Topic名称和Message Key进行模糊查询, 得到符合条件的消息的信息列表。

使用场景

使用本接口根据Message Key查询消息的方式属于模糊查询。由于业务方的Key可能不唯一, 所以查询结果可能为多条。

本接口用在业务方无法获取Message ID的场景, 在该场景下的调用步骤如下所示:

1. 调用本接口模糊查询得到Message ID列表。
2. 调用OnsMessageGetByMsgId接口精确查询。关于OnsMessageGetByMsgId接口的详细信息, 请参见[按Message ID查询消息 \(/json/message/getById\)](#)。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询的消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间, 则无需填写该参数。 如果您的实例有命名空间, 则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需查询的消息的Topic
key	String	是	需查询的消息的Message Key

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	List(OnsRestMessageDo)	查询结果

OnsRestMessageDo数据结构列表

名称	类型	描述
topic	String	消息的Topic
propertyList	List(MessageProperty)	消息属性列表
body	String	消息体
storeSize	Integer	消息大小
bornTimestamp	Long	生成时间戳
bornHost	String	生成该消息的客户端实例
storeTimestamp	Long	被Broker存储的时间戳
storeHost	String	存储该消息的服务器实例
msgId	String	消息ID, 即Message ID
bodyCRC	Integer	消息体CRC校验值
reconsumeTimes	Integer	消息重试消费的次数

2.6.7.4. 根据Topic查询消息

(/json/message/getByPagedTopic)

通过传入Topic名称和时间段, 分页查询指定时间段内该Topic内存在的所有消息。

使用场景

按Topic查询消息的方式属于范围查询, 查询结果会通过分页的方式展示给调用方, 仅适用于没有任何搜索条件 (Message ID 和 Message Key) 查询场景。需注意的是, 根据Topic查询消息应尽可能缩短查询的时间区间, 否则匹配的消息过多, 将无法定位问题。

调用流程如下所示:

1. 传入Topic、起止时间以及每页的大小, 进行分页查询。如果有消息, 默认返回第一页的消息、总页数和查询任务ID。
2. 根据返回的结果中的查询任务ID, 进行后续取消息, 取消息时传入查询任务ID以及当前页数。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	需查询的消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需查询的消息的Topic名称
beginTime	Long	是	查询范围的起始时间戳
endTime	Long	是	查询范围的终止时间戳
taskId	String	否	查询任务的ID，首次查询不需要输入，后续取消息必须传入，根据前一次的返回结果取出该字段
currentPage	Integer	是	当前取第几页消息，从1开始递增，最大值为50，取消息时不可超过最大页数
pageSize	Integer	否	分页查询，每页最多显示消息数量，默认是20，最小5条，最多50条

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
MsgFoundDo	MsgFoundDo	查询结果

MsgFoundDo 数据结构列表

名称	类型	描述
taskId	String	第一次新建查询时，返回的查询任务ID，用于后续取消息
maxPageCount	Long	查询结果最大页数
currentPage	Long	当前位于第几页
data	List(OnsRestMessageDo)	当前页的消息集合，该类型与messageGetById的返回结果一致


2.6.8. 发布订阅统计

2.6.8.1. Topic写入统计 (/json/trend/topicInputTps)

查询一段时间内指定的Topic的消息写入报表数据。

使用场景

您可以使用本接口获取在线上环境统计的Topic的消息数据信息，可以根据需要择查询指定时间段内Topic的写入量或者TPS曲线数据。

 **注意** 如果应用方消息量比较小，间隔不均匀，此时采用TPS查询可能会导致数据不明显，建议使用总量查询。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
instanceId	String	是	指定的Topic所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需查询的Topic名称
beginTime	Long	是	查询区间起始毫秒时间戳
endTime	Long	是	查询区间终止毫秒时间戳
period	Long	是	采样周期，单位分钟，取值可为 1、5、10
type	Integer	是	查询的类型。取值说明如下： <ul style="list-style-type: none"> 0: 查询指定时间段内Topic的写入量 1: 查询指定时间段内Topic的TPS曲线数据

返回参数列表

名称	类型	描述
data	Data	数据集合

Data数据集定义列表

名称	类型	描述
title	String	Table的名称
records	List(StatsDataDo)	采集点信息

StatsDataDo数据集定义列表

名称	类型	描述
x	Long	横轴，毫秒时间戳
y	Float	纵轴，数据（TPS或者总量）

2.6.8.2. Group ID投递统计


(/json/trend/groupOutputTps)

查询指定的Group ID在一段时间内消费消息的统计信息。

使用场景

线上环境生成消费数据报表时，可以使用本接口查询指定的Group ID在一段时间内的以下统计信息：

- 消费消息总量的统计曲线
- 消费消息TPS的统计曲线

 **注意** 如果应用方消息量比较小，间隔不均匀，此时采用TPS查询可能会导致数据不明显，建议查询消费消息总量。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
instanceId	String	是	需统计数据的Group ID所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
topic	String	是	需要查询的Topic名称
groupId	String	是	需要查询的Group ID
beginTime	Long	是	查询区间起始毫秒时间戳
endTime	Long	是	查询区间终止毫秒时间戳
period	Long	是	采样周期，单位分钟，取值可为 1、5、10
type	Integer	是	查询的类型。取值说明如下： <ul style="list-style-type: none"> 0：查询消费消息总量的统计曲线 1：查询消费消息 TPS 的统计曲线

返回参数列表

其他参数请参考[公共返回值解析](#)。

名称	类型	描述
data	Data	数据集合

Data数据集定义列表

名称	类型	描述
title	String	Table的名称
records	List(StatsDataDo)	采集点信息

StatsDataDo数据集定义列表

名称	类型	描述
x	Long	横轴，毫秒时间戳

名称	类型	描述
y	Float	纵轴，数据（TPS或者总量）

2.6.9. 授权管理

2.6.9.1. 创建主账号授权（/json/empower/create）

调用该接口可以将目标Topic授权给指定的其他主账号使用，包括发布消息和订阅消息的权限。

使用场景

RocketMQ的Topic资源属于申请的Owner账号所有，当需要跨账号访问Topic资源时，Owner账号可以使用授权接口对目标主账号进行发布消息或者接收消息的权限授予。

说明

- 授权可以支持三种权限：发布，订阅，发布&订阅。
- 发起授权的账号必须是资源的Owner账号，目标账号是主账号。
- 授权后，目标账号仍然需要创建自己的订阅GID，而不是使用Owner账号的GID来收消息。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	实例ID
empowerUser	Long	是	授权的目标账号的User ID
topic	String	是	授权的目标Topic，该Topic必须是当前用户所有
relation	Integer	是	授权类型，取值说明如下： <ul style="list-style-type: none"> • 2：代表授权发送 • 4：代表授权订阅 • 6：代表授权发送和订阅

返回参数列表

返回data为null。

2.6.9.2. 创建子账号授权（/json/empower/createBySub）

调用该接口可以将目标Topic授权给指定子账号使用，包括发布消息和订阅消息的权限。

使用场景

RocketMQ的Topic资源属于申请的Owner账号所有，当需要跨账号访问Topic资源时，Owner账号可以使用子账号授权接口对该用户的子账号进行发布消息或者接收消息的权限授予。

说明

- 授权可以支持三种权限：发布，订阅，发布&订阅。
- 发起授权的账号必须是资源的Owner账号，目标账号是该账号的子账号。
- 授权后，目标账号仍然需要创建自己的订阅GID，而不是使用Owner账号的GID来收消息。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	实例ID
empowerUser	Long	是	授权的目标子账号的名称
topic	String	是	授权的目标Topic，该Topic必须是当前用户所有
relation	Integer	是	授权类型，取值说明如下： <ul style="list-style-type: none"> • 2：代表授权发送 • 4：代表授权订阅 • 6：代表授权发送和订阅

返回参数列表

返回data为null。

2.6.9.3. 删除授权 (/json/empower/delete)

调用该接口可以对已经创建的授权关系进行删除，以解除之前创建的跨账号访问资源的权限。

使用场景

在创建授权后，如果需要解除跨账号的Topic访问权限，Owner账号可以调用该接口解除。

说明

- 调用方必须是资源的Owner账号。
- 需要删除的授权信息必须存在。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	实例ID
empowerUser	Long	是	授权的目标用户User ID，支持子账号ID

名称	类型	是否必须	描述
topic	String	是	授权的目标Topic, 该Topic必须是当前用户所有

返回参数列表

返回data为null。

2.6.9.4. 查询授权列表 (/json/empower/list)

调用该接口查询目标Topic的授权列表。

使用场景

Topic的所有者查询自己名下的Topic是否对特定账号授权, 授权的类型可以通过该接口查询。

请求参数列表

其他参数请参考[公共参数设置](#)。

名称	类型	是否必须	描述
instanceId	String	是	实例ID
empowerUser	Long	是	授权的目标用户User ID, 支持子账号ID
topic	String	是	授权的目标Topic, 该Topic必须是当前用户所有

返回参数列表

以JSON的格式返回该Topic的授权列表, 包含授权对象和授权类型等信息。

2.6.10. 死信队列管理

2.6.10.1. 按Message ID查询死信消息

(/json/message/getByIdInDLQ)

通过传入Message ID查询指定的死信消息。查询到的信息包括死信消息的存储时间、消息体、Key和Tag等属性。

描述

根据Message ID查询死信消息的方式属于精确查询, 查询条件需要的Message ID从消息发送成功的SendResult中获取, 或者从批量查询的接口中获取。

请求参数列表

名称	类型	是否必需	描述
----	----	------	----

名称	类型	是否必需	描述
InstanceId	String	是	需查询消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
GroupId	String	是	需查询的消费端Group ID
MsgId	String	是	需查询的消息的ID，即Message ID

返回参数列表

Data数据结构列表

名称	类型	描述
Topic	String	消息的Topic
PropertyList	List(MessageProperty)	消息属性列表
Body	String	消息体
StoreSize	Integer	消息大小
BornTimestamp	Long	生成时间戳
BornHost	String	生成该消息的客户端实例
StoreTimestamp	Long	被服务端存储的时间戳
StoreHost	String	存储该消息的服务器实例
MsgId	String	消息ID，即Message ID
BodyCRC	Integer	消息体CRC校验值
ReconsumeTimes	Integer	消息重试消费的次数

2.6.10.2. 按Group ID查询死信消息

(/json/message/getByPagedDLQ)

通过传入Group ID和时间段，分页查询指定时间段内该Group ID内存在的所有死信消息。

描述

按Group ID查询死信消息的方式属于范围查询，查询结果会通过分页的方式展示给调用方，仅适用于没有任何搜索条件Message ID查询场景。需注意的是，根据Group ID查询死信消息应尽可能缩短查询的时间区间，否则匹配的消息过多，不利于定位问题。

调用流程如下所示：

1. 传入Group ID、起止时间以及每页的大小，进行分页查询。如果有消息，默认返回第一页的消息、总页数和查询任务ID。
2. 根据返回的结果中的查询任务ID，进行后续取消息，取消息时传入查询任务ID以及当前页数。

请求参数列表

名称	类型	是否必需	描述
Instanceld	String	是	需查询的消息所对应的实例ID。 <ul style="list-style-type: none"> 如果您的实例无命名空间，则无需填写该参数。 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
Groupld	String	是	需查询的消息所对应的实例ID
BeginTime	Long	是	查询范围的起始时间戳
EndTime	Long	是	查询范围的终止时间戳
Taskld	String	否	查询任务的ID，首次查询不需要输入，后续取消息必须传入，根据前一次的返回结果取出该字段
CurrentPage	Integer	是	当前取第几页消息，从1开始递增，最大值为50，取消息时不可超过最大页数
PageSize	Integer	否	分页查询，每页最多显示消息数量，默认是20，最小5条，最多50条

返回参数列表

名称	类型	描述
MsgFoundDo	MsgFoundDo	查询结果

MsgFoundDo数据结构列表

名称	类型	描述
TaskId	String	第一次新建查询时，返回的查询任务ID，用于后续取消息
MaxPageCount	Long	查询结果最大页数
CurrentPage	Long	当前位于第几页
Data	List(OnsRestMessageDo)	当前页的消息集合，该类型与OnsDLQMessageGetById的返回结果一致

2.6.10.3. 重发死信消息

(/json/message/resendByIdInDLQ)

通过传入Message ID重发指定的死信消息，使该消息能够被Consumer再次消费。

描述

本接口让死信消息复活，让消息能够被Consumer再消费一次。

死信消息已达最大重试次数，若重发后再次消费失败，则死信队列中将增加一条Message ID相同的消息，可通过控制台死信队列的查询页面或者管控API查看该结果。您可通过同一Message ID的消息出现次数判断重发后依然消费失败的次数。

注意

- 由于死信消息是经过重试逻辑依然消费失败的消息，通常意味着消费逻辑可能存在问题，故请先查明多次消费失败的原因，再调用本接口重新发送。
- 当前RocketMQ不管理死信重发的状态，重发后消费成功的消息不会从死信队列中移除。需自行管理、判断消息是否需要重发。避免多次无意义地重发，从而造成重复消费。

请求参数列表

名称	类型	是否必需	描述
InstanceId	String	是	需查询消息所对应的实例ID。 <ul style="list-style-type: none"> • 如果您的实例无命名空间，则无需填写该参数。 • 如果您的实例有命名空间，则该参数为必填。 实例命名空间详情请参见 产品更新说明 。
MsgId	String	是	需查询的消息的ID，即Message ID

名称	类型	是否必需	描述
Groupid	String	是	需查询的消费端Group ID

返回参数列表

详情请参见[公共返回值解析](#)。

3.SDK参考

3.1. 概述

消息队列RocketMQ版（简称RocketMQ）是分布式系统中重要的组件，主要解决系统解耦、异步消息、削峰填谷等问题，实现高性能、高可用、可伸缩和最终一致性架构。

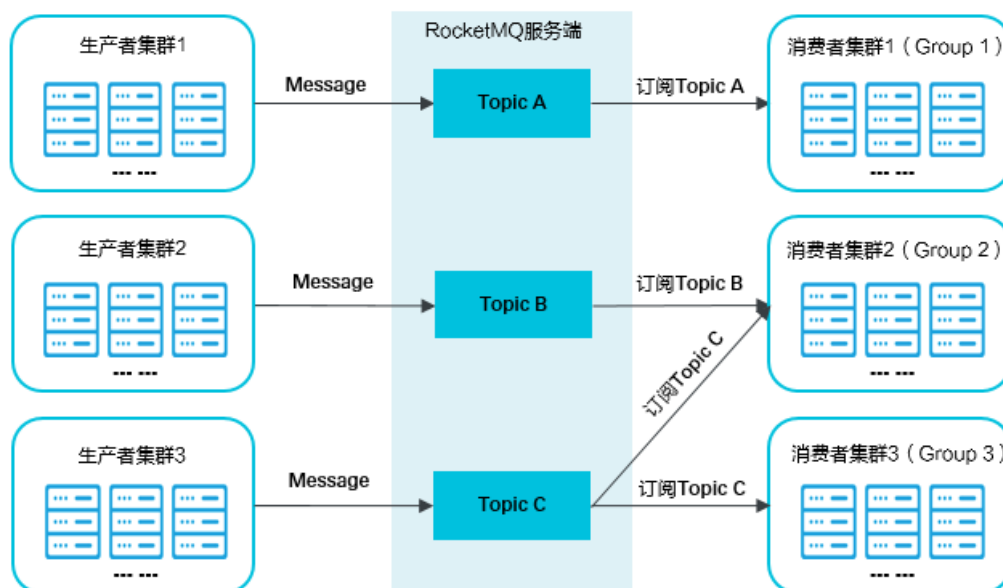
本参考规定RocketMQ应用节点与消息队列服务中心间数据传输服务管理方法，包括服务遵循的规范，以及服务管理相关的公共API。

本参考适用于使用RocketMQ Java SDK，基于TCP协议进行消息传输的场景。

3.2. 消息模型

RocketMQ支持发布/订阅模型，发布者可以将一条消息发送至消息服务端，多个接收方进行订阅。如下图：

Pub-Sub模型



3.3. 产品更新说明

本文介绍消息队列RocketMQ版（简称RocketMQ）从V3.8.0到V3.8.1的产品更新说明，帮助您更快地上手更新后的版本。

资源隔离优化（实例化支持）

RocketMQ开放实例化功能，解决多租之间的隔离问题，每个用户可以购买多个实例，每个实例之间逻辑隔离。

为兼容老用户的既有资源，RocketMQ实例的命名空间分为两种：

- 默认实例（兼容老用户的既有资源）
 - 无独立命名空间，实例内或者跨实例之间，资源命名必须全局唯一。
 - 对RocketMQ老用户来说，既有资源将自动生成一个默认的无独立命名空间实例；无既有资源，则可选择创建不带命名空间的实例，最多创建1个。

- 接入点配置（可在控制台实例详情页面中获取）。

```
// 推荐配置
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxxx");
// 兼容配置（不推荐继续使用, 建议逐渐升级为推荐配置）
properties.put(PropertyKeyConst.ONSAddr, "xxxx");
```

- 新建实例
 - 有独立命名空间，资源命名确保实例内唯一，跨实例之间可重名。
 - 接入点配置（可在控制台实例详情页面中获取）。

```
// 推荐配置
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
```

- ons-client 客户端必须升级到最新版本：
 - JAVA: 1.8.7.1.Final
 - C/C++: v2.0.0
 - .NET: 1.1.3

资源申请流程优化

原RocketMQ的资源包括3个部分：消息主题（Topic）、生产者（Producer ID）、消费者（Consumer ID），三者之间都是多对多的对应关系，三者之间的关系复杂而难以理解；对于中、大型企业客户来说，每创建一个Topic，都要关联或者更新Producer ID和Consumer ID，流程过于复杂。

因此，为进一步优化用户体验，降低新用户的接入门槛，对资源申请的接入流程进行简化。

资源申请方面，主要包括以下部分：

- Topic管理（保持不变）
 - Topic资源申请，一级消息类型，通过Topic对消息进行分类。
- Group管理
 - 取消原Producer ID的申请，与原Consumer ID统一为Group ID，即对应控制台上，取消原生产者管理，与原消费者管理统一为Group管理。
 - 取消原申请Producer ID或Consumer ID时需要指定Topic的关联约束，只需申请Group ID即可在代码中直接配置与Topic的关联使用。
 - 兼容性相关问题
 - 原申请的Producer ID列表将不再显示，不影响当前业务的正常使用。
 - 原申请的以“CID-”或者“CID_”打头的Consumer ID仍可以正常使用，代码中既可在PropertyKeyConst.ConsumerId中设置，亦可在PropertyKeyConst.GROUP_ID中设置。
- 示例代码

说明

- 建议您将ons-client客户端升级为最新版本：
 - JAVA: 1.8.7.1.Final
 - C/C++: v2.0.0
 - .NET: 1.1.3
- 原来已经在使用的Producer ID或者Consumer ID保持兼容，不影响当前在线业务的正常使用，但建议逐步替换为新的推荐配置方式。

- 推荐方式：将Producer ID、Consumer ID的概念融合成Group ID。

```
// 设置PropertyKeyConst.GROUP_ID, 原PropertyKeyConst.ProducerId和PropertyKeyConst.ConsumerId会被@Deprecated
properties.put(PropertyKeyConst.GROUP_ID, "原CID-XXX或GID-XXX");
```

- 兼容方式：Producer ID用来标识生产者，Consumer ID用来标识消费者。

```
// 创建Producer时需要配置PropertyKeyConst.ProducerId。
properties.put(PropertyKeyConst.ProducerId, "原PID-XXX或GID-XXX");
// 创建Consumer时需要配置PropertyKeyConst.ConsumerId。
properties.put(PropertyKeyConst.ConsumerId, "原CID-XXX或GID-XXX");
```

3.4. 传输协议与字符编码

RocketMQ SDK底层通过TCP协议提供服务，用户请求和响应的字符编码都是采用UTF-8编码。

传输层加密

RocketMQ提供对传输层安全性协议（TLS）的支持，您可以通过SDK的配置来决定是否使用加密的传输层链接，但前提是得到服务端的支持。

配置方式：在启动客户端程序时，设置JVM属性来决定是否使用加密的传输层链接。启动命令为`java -Dtls.enable=true`。默认情况下，TLS链接不开启，设置为true将启用加密的传输层链接。

3.5. 常用类及其方法

3.5.1. 常量类PropertyKeyConst

该类字段主要包括以下内容。

名称	类型	默认值	描述
AccessKey	String	-	您在阿里云账号管理控制台中创建的AccessKey ID，用于身份认证。
SecretKey	String	-	您在阿里云账号管理控制台中创建的AccessKey Secret，用于身份认证。
OnsChannel	String	ALIYUN	用户渠道，阿里云：ALIYUN，聚石塔用户：CLOUD。

名称	类型	默认值	描述
NAMESRV_ADDR	String	-	设置RocketMQ TCP协议接入点。
GROUP_ID	String	-	标识一类Producer或Consumer，您在RocketMQ控制台创建的Group ID。
MessageModel	String	CLUSTERING	设置Consumer实例的消费模式，集群消费：CLUSTERING，广播消费：BROADCASTING。
ConsumeThreadNums	String	64	消费线程数量。
MaxReconsumeTimes	String	16	设置消息消费失败的最大重试次数。
ConsumeTimeout	String	15	设置每条消息消费的最大超时时间，超过设置时间则被视为消费失败，等下次重新投递再次消费。每个业务需要设置一个合理的值，单位（分钟）。
ConsumeMessageBatchMaxSize	String	1	BatchConsumer每次批量消费的最大消息数量，默认值为1，允许自定义范围为[1, 32]，实际消费数量可能小于该值。
CheckImmunityTimeInSecs	String	30	设置事务消息第一次回查的最快时间，单位（秒）。
shardingKey	String	-	顺序消息中用来计算不同分区的值。
suspendTimeMillis	String	3000	只适用于顺序消息，设置消息消费失败的重试间隔时间，单位（毫秒）。

3.5.2. 工厂类ONSFactory

3.5.2.1. createConsumer

- 语法：static Consumer createConsumer(Properties properties)
- 描述：创建RocketMQ普通消息、定时和延时消息、分布式事务消息等消息类型的消费者实例。
- 参数：

名称	类型	描述
properties	Properties	<p>详情参考PropertyKeyConst说明。</p> <p>必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.ConsumerId、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey。</p> <p>可选属性：PropertyKeyConst.ConsumeThreadNums、PropertyKeyConst.ConsumeTimeout。</p>

- 返回值

类型	描述
Consumer	RocketMQ普通消息、定时和延时消息、分布式事务消息等消息类型的消费者实例。

3.5.2.2. createBatchConsumer

- 语法：static BatchConsumer createBatchConsumer(Properties properties)
- 描述：创建RocketMQ批量消费者实例。
- 参数：

名称	类型	描述
properties	Properties	<p>详情参考PropertyKeyConst说明；</p> <p>必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.GROUP_ID、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey；</p> <p>可选属性：PropertyKeyConst.ConsumeThreadNums、PropertyKeyConst.ConsumeTimeout、PropertyKeyConst.ConsumeMessageBatchMaxSize。</p>

- 返回值：

类型	描述
BatchConsumer	BatchConsumer

3.5.2.3. createOrderedConsumer

- 语法：static OrderConsumer createOrderedConsumer(Properties properties)
- 描述：创建RocketMQ顺序消息消费者。
- 参数：

名称	类型	描述
properties	properties	<p>详情参考PropertyKeyConst说明；</p> <p>必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.GROUP_ID、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey；</p> <p>可选属性：PropertyKeyConst.ConsumeThreadNums、PropertyKeyConst.ConsumeTimeout。</p>

- 返回值：

类型	描述
OrderConsumer	RocketMQ顺序消息消费者实例

3.5.2.4. createOrderProducer

- 语法：static OrderProducer createOrderProducer(Properties properties)
- 描述：创建RocketMQ顺序消息发布者。
- 参数：

名称	类型	描述
properties	Properties	详情参考PropertyKeyConst说明； 必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.GROUP_ID、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey、PropertyKeyConst.ShardingKey； 可选属性：PropertyKeyConst.SendMsgTimeoutMillis

- 返回值：

类型	描述
OrderProducer	RocketMQ顺序消息发布者实例

3.5.2.5. createProducer

- 语法：static Producer createProducer(Properties properties)
- 描述：创建RocketMQ普通消息、定时和延时消息发布者。
- 参数：

名称	类型	描述
properties	Properties	详情参考PropertyKeyConst说明。 必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.GROUP_ID、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey。 可选属性：PropertyKeyConst.SendMsgTimeoutMillis。

- 返回值：

类型	描述
Producer	RocketMQ普通消息、定时和延时消息发布者实例。

3.5.2.6. createTransactionProducer

- 语法：static TransactionProducer createTransactionProducer(Properties properties, LocalTransactionChecker checker)
- 描述：创建RocketMQ分布式事务消息发布者。
- 参数：

名称	类型	描述
properties	Properties	详情参考PropertyKeyConst说明； 必填属性：PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.GROUP_ID、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey； 可选属性：PropertyKeyConst.SendMsgTimeoutMillis、PropertyKeyConst.CheckImmunityTimeInSeconds。
checker	LocalTransactionChecker	详情参考LocalTransactionChecker说明。

- 返回值：

类型	描述
TransactionProducer	RocketMQ分布式事务消息发布者实例

3.5.3. 消息类Message

3.5.3.1. 消息类Message概述

消息类，表示要发送的消息。一条合法的消息，应包含主题(Topic)，消息体以及可选的标签和业务标识。

注意：我们对每条消息的自定义键值对的长度没有限制，但所有的自定义键值对，系统键值对序列化后，所占空间不能超过32767字节。

3.5.3.2. Message默认构造方法

- 语法：Message()
- 描述：默认构造函数，创建消息对象。
- 参数：无

3.5.3.3. Message有参构造方法（一）

- 语法：public Message(String topic, String tag, String key, byte body)
- 描述：有参构造函数，创建消息对象。
- 参数：

名称	类型	描述
topic	String	消息主题，最长不超过255个字符；由a-z, A-Z, 0-9, 以及中划线"-"和下划线"_"构成。
tag	String	消息标签，请使用合法标识符，尽量简短且见名知意。
key	String	业务标识。
body	byte	消息体，消息体长度默认不超过4 MB（包含消息属性，其中消息属性不超过8 KB），具体请参阅集群部署文档描述。

3.5.3.4. Message有参构造方法（二）

- 语法：public Message(String topic, String tags, byte body)
- 描述：有参构造函数，创建消息对象。
- 参数：

名称	类型	描述
topic	String	消息主题，最长不超过255个字符；由a-z, A-Z, 0-9, 以及中划线"-"和下划线"_"构成。
tags	String	消息标签，请使用合法标识符，尽量简短且见名知意。
body	byte	消息体，消息体长度默认不超过4 MB（包含消息属性，其中消息属性不超过8 KB），具体请参阅集群部署文档描述。

3.5.3.5. putUserProperties

- 语法：public void putUserProperties(String key, String value)
- 描述：添加用户自定义属性键值对；该键值对在消费消息时可被获取。
- 参数：

名称	类型	描述
key	String	自定义属性键。
value	String	自定义属性值。

- 返回值：无

3.5.3.6. getUserProperties

- 语法：public String getUserProperties(String key)
- 描述：获取用户自定义属性的值。
- 参数：

名称	类型	描述
key	String	自定义属性键。

- 返回值:

类型	描述
String	用户自定义属性值

3.5.3.7. getTopic

- 语法: `public String getTopic()`
- 描述: 获取消息的主题。
- 参数: 无
- 返回值:

类型	描述
String	消息主题

3.5.3.8. setTopic

- 语法: `public void setTopic(String topic)`
- 描述: 设置消息主题。
- 参数:

名称	类型	描述
topic	String	消息主题

- 返回值: 无

3.5.3.9. getTag

- 语法: `public String getTag()`
- 描述: 获取消息标签。
- 参数: 无
- 返回值:

类型	描述
String	消息标签

3.5.3.10. setTag

- 语法: `public void setTag(String tag)`

- 描述：设置消息标签。
- 参数：

名称	类型	描述
tag	String	消息标签

- 返回值：无

3.5.3.11. getKey

- 描述：获取业务标识。
- 语法：public String getKey()
- 参数：无
- 返回值：

类型	描述
String	业务标识

3.5.3.12. setKey

- 语法：public void setKey(String key)
- 描述：设置业务标识。
- 参数：

名称	类型	描述
key	String	业务标识

- 返回值：无

3.5.3.13. getMsgID

- 语法：public String getMsgID()
- 描述：获取消息ID。
- 参数：无
- 返回值：

类型	描述
String	消息ID

3.5.3.14. setMsgID

- 语法：public void setMsgID(String msgid)

- 描述：设置消息ID
- 参数：

名称	类型	描述
msgid	String	消息ID

- 返回值：无

3.5.3.15. getReconsumeTimes

- 语法：public int getReconsumeTimes()
- 描述：消息消费时，获取消息已经被重试消费的次数。
- 参数：无
- 返回值：

类型	描述
String	重试消费次数

3.5.3.16. setReconsumeTimes

- 语法：public void setReconsumeTimes(int value)
- 描述：设置消息重试消费次数。
- 参数：

名称	类型	描述
value	int	重试消费次数

- 返回值：无

3.5.3.17. getBornTimestamp

- 语法：public long getBornTimestamp()
- 描述：获取消息的生产时间。
- 参数：无
- 返回值：

类型	描述
String	消息的生产时间

3.5.3.18. setBornTimestamp

- 语法：public void setBornTimestamp(long value)
- 描述：设置消息的产生时间。

- 参数：

名称	类型	描述
value	long	消息生产时间。

- 返回值：无

3.5.3.19. getBornHost

- 语法：public String getBornHost()
- 描述：获取产生消息的主机。
- 参数：无
- 返回值：

类型	描述
String	产生消息的主机

3.5.3.20. setBornHost

- 语法：public void setBornHost(String value)
- 描述：设置生产消息的主机。
- 参数：

名称	类型	描述
value	long	生产消息的主机。

- 返回值：无

3.5.3.21. getStartDeliverTime

- 语法：public long getStartDeliverTime()
- 描述：获取定时消息开始投递时间。
- 参数：无
- 返回值：

类型	描述
String	定时消息开始投递时间

3.5.3.22. setStartDeliverTime

- 语法：public void setStartDeliverTime(long value)
- 描述：设置消息的定时投递时间（绝对时间），最大延迟时间为40天。
- 参数：

名称	类型	描述
value	long	定时消息的开始投递时间。

- 返回值：无

3.5.3.23. getShardingKey

- 语法：public long getShardingKey()
- 描述：获取顺序消息中用来计算不同分区的值。
- 参数：无
- 返回值：

类型	描述
String	顺序消息中用来计算不同分区的值

3.5.3.24. setShardingKey

- 语法：public void setShardingKey(String value)
- 描述：设置顺序消息中用来计算不同分区的值。
- 参数：

名称	类型	描述
String	long	顺序消息中用来计算不同分区的值。

- 返回值：无

3.5.4. 异常类ONSClientException

3.5.4.1. ONSClientException默认构造方法

- 语法：public ONSClientException()
- 描述：默认异常构造函数。
- 参数：无

3.5.4.2. ONSClientException有参构造方法（一）

- 语法：public ONSClientException(String message)
- 描述：异常接口构造函数。
- 参数：

名称	类型	描述
message	String	需要向外传递的异常信息

3.5.4.3. ONSClientException有参构造方法（二）

- 语法：public ONSClientException(Throwable cause)
- 描述：异常接口构造函数。
- 参数：

名称	类型	描述
message	String	需要向外传递的异常信息

3.5.4.4. ONSClientException有参构造方法（三）

- 语法：public ONSClientException(String message, Throwable cause)
- 描述：异常接口构造函数。
- 参数：

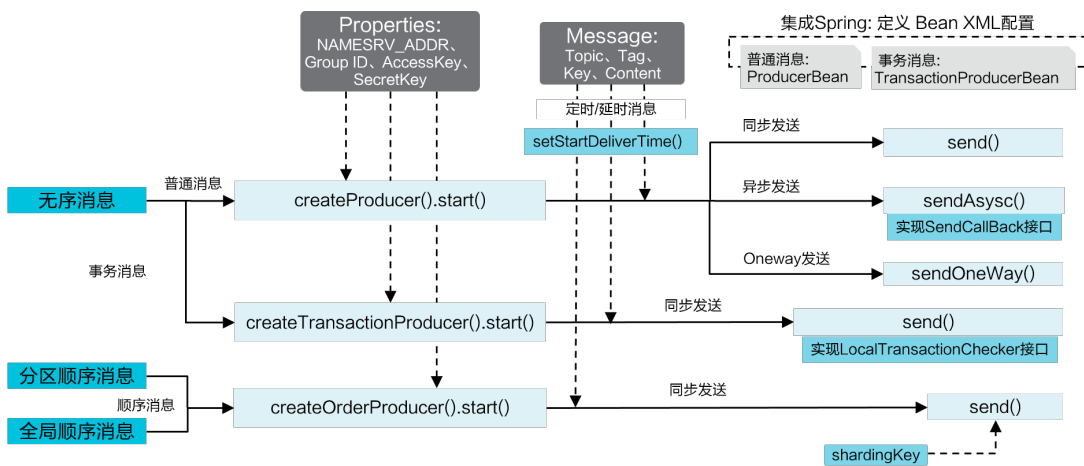
名称	类型	描述
message	String	需要向外传递的异常信息
cause	Throwable	需要向外传递的异常信息

3.5.5. 发送消息接口

3.5.5.1. 发送消息接口概述

本单元主要详述RocketMQ发送各种类消息的接口规范。不同消息类型与各个接口、参数的对应关系如下图所示。

发布消息相关接口参数说明



3.5.5.2. 发送结果SendResult

3.5.5.2.1. SendResult默认构造方法

- 语法：public SendResult()
- 描述：默认构造函数，创建消息发送结果对象。
- 参数：无

3.5.5.2.2. getMessageId

- 语法：public long getMessageId()
- 描述：获取被发送消息的消息 ID。
- 参数：无
- 返回值：

类型	描述
String	被发送消息的消息 ID

3.5.5.2.3. getTopic

- 语法：public String getTopic()
- 描述：获取被发送消息的消息主题。
- 参数：无
- 返回值：

类型	描述
String	被发送消息的消息主题

3.5.5.2.4. toString

- 语法：public String toString()
- 描述：序列化发送结果。
- 参数：无
- 返回值：

类型	描述
String	发送结果的String表示

3.5.5.3. 生产者实例Producer

3.5.5.3.1. 生产者实例Producer概述

发布者对象，可以通过该对象接口发送普通消息，发送定时/延时消息。

- 线程安全性：本接口的实现类都是线程安全的，可以多线并发发送消息。
- 一个实例可以发送多个主题的消息。
- 正常情况下，一个实例足够高效完成本模块的发送任务，无需创建多个实例

RocketMQ普通消息支持3种发送方式：可靠同步发送、可靠异步发送、单向（Oneway）发送。

3.5.5.3.2. send

- 语法：public SendResult send(Message message)
- 描述：同步发送消息，只要不抛异常就表示成功；支持普通消息，发送定时/延时消息。
- 参数：

名称	类型	描述
message	Message	要发送的消息对象

- 返回值：

类型	描述
SendResult	发送结果，含消息Id，消息主题等

3.5.5.3.3. sendOneway

- 语法：public void sendOneway(Message message)
- 描述：发送消息，Oneway形式，服务器不应答，无法保证消息是否成功到达服务器；支持普通消息，发送定时/延时消息。
- 参数：

名称	类型	描述
message	Message	要发送的消息对象

- 返回值：无

3.5.5.3.4. sendAsync

- 语法：public void sendAsync(Message message, SendCallback sendCallback)
- 描述：发送消息，异步Callback形式；支持普通消息，发送定时/延时消息。
- 参数：

名称	类型	描述
message	Message	要发送的消息对象
sendCallback	SendCallback	发送完成要执行的回调函数

- 返回值：无

3.5.5.3.5. isStarted

- 语法：public boolean isStarted()

- 描述：检查服务是否已经启动。
- 参数：无
- 返回值：

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.5.3.6. isClosed

- 语法：public boolean isClosed()
- 描述：检查服务是否已经关闭。
- 参数：无
- 返回值：

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.5.3.7. start

- 语法：public void start()
- 描述：启动消息发送 Producer 服务。
- 参数：无
- 返回：无

3.5.5.3.8. shutdown

- 语法：public void shutdown()
- 描述：关闭消息发送 Producer 服务。
- 参数：无
- 返回：无

3.5.5.4. 分布式事务消息生产者实例 TransactionProducer

3.5.5.4.1. send

- 语法：SendResult send(Message message, LocalTransactionExecuter executer, Object arg)
- 描述：该方法用来发送一条事务型消息。事务型消息发送分为三个步骤：
 - 本服务实现类首先发送一条半事务消息到消息服务器；
 - 通过executer执行本地事务；
 - 根据上一步骤执行结果, 决定发送提交或者回滚第一步发送的半事务消息；
- 参数：

名称	类型	描述
message	Message	要发送的事务型消息
executer	LocalTransactionExecuter	本地事务执行器, 详情请参考 LocalTransactionExecuter类说明
arg	Object	应用自定义参数, 该参数可以传入本地事务执行器

- 返回值:

类型	描述
SendResult	返回发送结果, 含消息Id, 消息主题等

3.5.5.4.2. isStarted

- 语法: public boolean isStarted()
- 描述: 检查服务是否已经启动。
- 参数: 无
- 返回值:

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.5.4.3. isClosed

- 语法: public boolean isClosed()
- 描述: 检查服务是否已经关闭。
- 参数: 无
- 返回值:

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.5.4.4. start

- 语法: public void start()
- 描述: 启动事务消息发送 Producer 服务。
- 参数: 无
- 返回值: 无

3.5.5.4.5. shutdown

- 语法：public void shut down()
- 描述：关闭事务消息发送 Producer 服务。
- 参数：无
- 返回值：无

3.5.5.4.6. 本地事务执行器LocalTransactionExecuter

3.5.5.4.6.1. execute

- 语法：TransactionStatus execute(Message msg, Object arg)
- 描述：执行本地事务，由应用来重写；
- 参数：

名称	类型	描述
message	Message	要发送的事务型消息
arg	Object	应用自定义参数，由send方法传入并回调

- 返回值：

类型	描述
TransactionStatus	返回事务执行结果，包括提交事务、回滚事务、未知状态

3.5.5.4.7. 分布式事务消息状态TransactionStatus

分布式事务半消息的状态，包括提交事务、回滚事务、未知状态。

枚举：

类型	描述
CommitTransaction	提交事务，对应的事务消息将投递给消费者
RollbackTransaction	回滚事务，对应的事务消息会被删除，不会投递给消费者
Unknow	未知状态，一般在用户无法确定事务是成功还是失败时使用，对于未知状态的事务，服务端会定期进行事务回查

3.5.5.5. 顺序消息生产者实例OrderProducer

3.5.5.5.1. send

- 语法：SendResult send(Message message, String shardingKey)
- 描述：发送顺序消息。
- 参数：

名称	类型	描述
message	shardingKey	要发送的顺序消息对象
shardingKey	String	顺序消息选择因子，发送方法基于 shardingKey 选择具体的消息队列

- 返回值：

类型	描述
SendResult	发送结果，含消息ID, 消息主题等

3.5.5.5.2. isStarted

- 语法：public boolean isStarted()
- 描述：检查服务是否已经启动。
- 参数：无
- 返回值：

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.5.5.3. isClosed

- 语法：public boolean isClosed()
- 描述：检查服务是否已经关闭。
- 参数：无
- 返回值：

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.5.5.4. start

- 语法：public void start()
- 描述：启动事务消息发送 Producer 服务。
- 参数：无
- 返回：无

3.5.5.5.5. shutdown

- 语法：public void shutdown()
- 描述：关闭事务消息发送 Producer 服务。

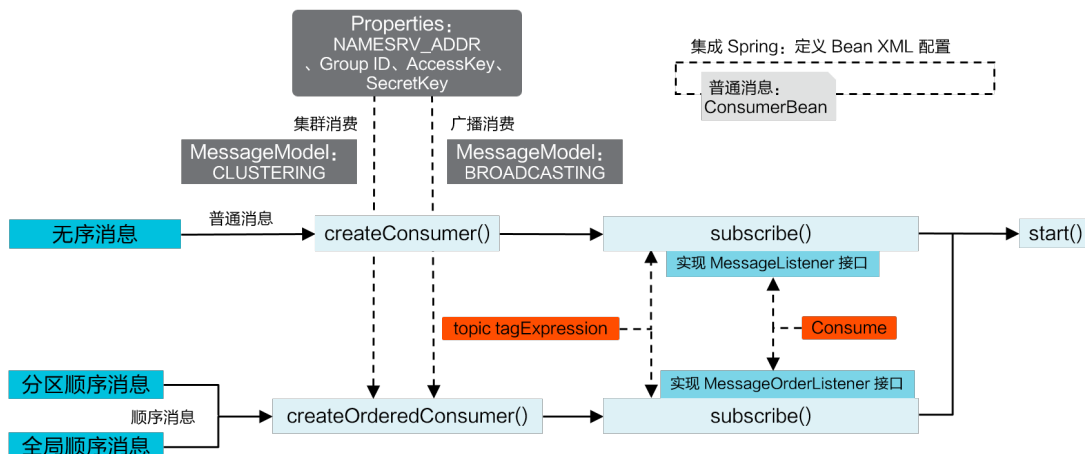
- 参数：无
- 返回：无

3.5.6. 订阅消息接口

3.5.6.1. 订阅消息接口概述

本节详述RocketMQ订阅各种类消息的接口规范。不同消息类型与各个接口、参数的对应关系如下图所示。

订阅消息接口相关参数说明



3.5.6.2. 消费者实例Consumer

3.5.6.2.1. 消费者实例Consumer概述

消息消费者实例，用来订阅消息，支持普通消息、事务消息、定时消息/延时消息。

3.5.6.2.2. subscribe

- 语法：public void subscribe(String topic, String subExpression, MessageListener listener)
- 描述：订阅消息，包括普通消息、事务消息、定时消息/延时消息。
- 参数：

名称	类型	描述
topic	String	消息主题
subExpression	String	订阅过滤表达式字符串，RocketMQ服务器依据此表达式进行过滤，只支持或 () 运算； 示例1: "tag1 tag2 tag3"; 示例2: subExpression 等于 null 或者 *，则表示全部订阅；
listener	MessageListener	消息回调监听器

- 返回值：无

3.5.6.2.3. unsubscribe

- 语法：void unsubscribe(String topic)
- 描述：取消某个topic订阅。
- 参数：

名称	类型	描述
topic	String	消息主题

- 返回值：无

3.5.6.2.4. isStarted

- 语法：public boolean isStarted()
- 描述：检查服务是否已经启动。
- 参数：无
- 返回值：

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.6.2.5. isClosed

- 语法：public boolean isClosed()
- 描述：检查服务是否已经关闭。
- 参数：无
- 返回值：

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.6.2.6. start

- 语法：public void start()
- 描述：启动消费者实例 Consumer 服务。
- 参数：无
- 返回值：无

3.5.6.2.7. shutdown

- 语法：public void shutdown()

- 描述：关闭消费者实例 Consumer 服务。
- 参数：无
- 返回值：无

3.5.6.3. 消息监听器MessageListener

3.5.6.3.1. 消费监听器MessageListener概述

消息监听器，Consumer注册消息监听器来订阅消息。

线程安全性要求：该接口会被Consumer的多个线程并发调用，用户需要保证并发安全性

3.5.6.3.2. consume

- 语法：Action consume(Message message, ConsumeContext context)
- 描述：消费消息接口，由应用来实现；网络抖动等不稳定的情形可能会带来消息重复，对重复消息敏感的业务可对消息做幂等处理。
- 参数：

名称	类型	描述
message	Message	消息对象
context	ConsumeContext	消费上下文，具体参考 ConsumeContext 说明

- 返回值：

类型	描述
Action	消费结果。 消费失败：如果应用抛出异常或者返回Null等价于返回Action.ReconsumeLater; 消费成功：返回 Action.CommitMessage。

3.5.6.4. 消息消费结果Action

3.5.6.4.1. 消息消费结果Action概述

消费消息的返回结果。

枚举：

名称	类型	描述
CommitMessage	String	消费成功，继续消费下一条消息
ReconsumeLater	String	消费失败，告知服务器稍后再投递这条消息，继续消费其他消息

3.5.6.4.2. valueOf

- 语法：public static Action valueOf(String name)
- 描述：返回带有指定名称的该类型的枚举常量，字符串必须与用于声明该类型的枚举常量的标识符完全匹配；（不允许有多余的空格字符）。
- 参数：

名称	类型	描述
name	String	要返回的枚举常量的名称

- 返回值：

类型	描述
Action	返回带有指定名称的枚举常量

- 异常：

类型	描述
IllegalArgumentException	如果该枚举类型没有带有指定名称的常量
NullPointerException	如果参数为空值

3.5.6.4.3. values

- 语法：public static Action[] values()
- 描述：按照声明该枚举类型常量的顺序, 返回包含这些常量的数组。
- 参数：无
- 返回值：

类型	描述
Action[]	按照声明该枚举类型的常量的顺序返回的包含这些常量的数组

3.5.6.5. 消费模式PropertyValueConst

消费模式常量类。

枚举：

名称	类型	描述
BROADCASTING	String	广播消费模式
CLUSTERING	String	集群消费模式

3.5.6.6. 顺序消费者实例OrderConsumer

3.5.6.6.1. 顺序消费者实例OrderConsumer概述

顺序消息消费者实例，通过改对象的接口可以先进先出的方式进行顺序订阅。

3.5.6.6.2. subscribe

- 语法：void subscribe(String topic, String subExpression, MessageOrderListener listener)
- 描述：订阅顺序消息。
- 参数：

名称	类型	描述
topic	String	消息主题
subExpression	String	订阅过滤表达式字符串，RocketMQ服务器依据此表达式进行过滤，只支持或 () 运算； 示例1: "tag1 tag2 tag3"; 实例2: subExpression 等于 null 或者 *，则表示全部订阅；
listener	MessageOrderListener	消息回调监听器

- 返回值：无

3.5.6.6.3. unsubscribe

- 语法：void unsubscribe(String topic)
- 描述：取消某个topic订阅。
- 参数：

名称	类型	描述
topic	String	消息主题

- 返回值：无

3.5.6.6.4. isStarted

- 语法：public boolean isStarted()
- 描述：检查服务是否已经启动。
- 参数：无
- 返回值：

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.6.6.5. isClosed

- 语法: public boolean isClosed()
- 描述: 检查服务是否已经关闭。
- 参数: 无
- 返回值:

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.6.6.6. start

- 语法: public void start()
- 描述: 启动消费者实例 Consumer 服务。
- 参数: 无
- 返回值: 无

3.5.6.6.7. shutdown

- 语法: public void shut down()
- 描述: 关闭消费者实例 Consumer 服务。
- 参数: 无
- 返回: 无

3.5.6.7. 消息监听器MessageOrderListener

3.5.6.7.1. 消息监听器MessageOrderListener概述

顺序消息监听器, Consumer注册消息监听器来订阅消息。

线程安全性要求: 该接口会被Consumer的多个线程并发调用, 用户需要保证并发安全性

3.5.6.7.2. consume

- 语法: OrderAction consume(Message message, ConsumeOrderContext context)
- 描述: 消费消息接口, 由应用来实现; 网络抖动等不稳定的情形可能会带来消息重复, 对重复消息敏感的业务可对消息做幂等处理。
- 参数:

名称	类型	描述
message	Message	消息对象
context	ConsumeOrderContext	消费上下文，具体参考 ConsumeOrderContext 说明；

- 返回值：

类型	描述
OrderAction	消费结果。 消费失败：如果应用抛出异常或者返回Null等价于返回 Action.Suspend； 消费成功：返回Action.Success。

3.5.6.8. 顺序消息消费结果OrderAction

3.5.6.8.1. 顺序消息消费结果OrderAction概述

顺序消息消费消息的返回结果。

枚举：

名称	类型	描述
Success	String	消费成功，继续消费下一条消息
Suspend	String	消费失败，挂起当前队列

3.5.6.8.2. valueOf

- 语法：public static OrderAction valueOf(String name)
- 描述：返回带有指定名称的该类型的枚举常量，字符串必须与用于声明该类型的枚举常量的标识符完全匹配；（不允许有多余的空格字符）。
- 参数：

名称	类型	描述
name	String	要返回的枚举常量的名称

- 返回值：

类型	描述
OrderAction	返回带有指定名称的枚举常量

- 异常：

类型	描述
IllegalArgumentException	如果该枚举类型没有带有指定名称的常量
NullPointerException	如果参数为空值

3.5.6.8.3. values

- 语法：public static OrderAction[] values()
- 描述：按照声明该枚举类型常量的顺序, 返回包含这些常量的数组。
- 参数：无
- 返回值：

类型	描述
OrderAction[]	按照声明该枚举类型的常量的顺序返回的包含这些常量的数组

3.5.6.9. 批量消费者实例BatchConsumer

3.5.6.9.1. 批量消费者实例BatchConsumer概述

批量消息消费者，用来通过批量的方式订阅消息。

3.5.6.9.2. subscribe

- 语法：void subscribe(String topic, String subExpression, BatchMessageListener listener)
- 描述：订阅消息，包括普通消息、事务消息、定时消息/延时消息。
- 参数：

名称	类型	描述
topic	String	消息主题
subExpression	String	订阅过滤表达式字符串，RocketMQ服务器依据此表达式进行过滤，只支持或 () 运算； 示例1: "tag1 tag2 tag3"; 实例2: subExpression 等于 null 或者 *，则表示全部订阅；
listener	BatchMessageListener	消息回调监听器

- 返回值：无

3.5.6.9.3. unsubscribe

- 语法：void unsubscribe(String topic)

- 描述：取消某个topic订阅。
- 参数：

名称	类型	描述
topic	String	消息主题

- 返回值：无

3.5.6.9.4. isStarted

- 语法：public boolean isStarted()
- 描述：检查服务是否已经启动。
- 参数：无
- 返回值：

类型	描述
boolean	如果检查服务已经启动返回 true; 其它情况返回 false

3.5.6.9.5. isClosed

- 语法：public boolean isClosed()
- 描述：检查服务是否已经关闭。
- 参数：无
- 返回值：

类型	描述
boolean	如果服务已关闭返回 true; 其它情况返回 false

3.5.6.9.6. start

- 语法：public void start()
- 描述：启动消费者实例 Consumer 服务。
- 参数：无
- 返回：无

3.5.6.9.7. shutdown

- 语法：public void shutdown()
- 描述：关闭消费者实例 Consumer 服务。
- 参数：无
- 返回：无

3.5.6.10. 批量消费消息监听器BatchMessageListener

3.5.6.10.1. 批量消费消息监听器BatchMessageListener概述

批量消息监听器，BatchConsumer注册消息监听器来批量地消费消息。

线程安全性要求：该接口会被Consumer的多个线程并发调用，用户需要保证并发安全性。

3.5.6.10.2. consume

- 语法：Action consume(Message message, ConsumeContext context)
- 描述：批量消费消息接口，由应用来实现需要注意网络抖动等不稳定的情形可能会带来消息重复，对重复消息敏感的业务可对消息做幂等处理。
- 参数：

名称	类型	描述
message	Message	消息对象
context	ConsumeContext	消费上下文，具体参考 ConsumeContext 说明

- 返回值：

类型	描述
Action	消费结果。 消费失败：如果应用抛出异常或者返回Null等价于返回Action.ReconsumeLater; 消费成功：返回 Action.CommitMessage;