阿里云
专有云Enterprise版
消息队列MQ专业版

开发指南

产品版本：V3.3.0

文档版本：20180312

为了无法计算的价值 | 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
	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。	**禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	**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。	单击 **确定**。
courier字体	命令。	执行 cd /d C:/windows 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	ipconfig *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	swich *{stand | slave}*

目录

法律声明.....I

通用约定.....I

1 API参考.....1

1.1 MQ Open API简介.....1

1.1.1 面向用户和场景.....1

1.1.2 适用场景和局限性.....1

1.1.3 Open API基本原理.....1

1.2 MQ Open API接入指南.....1

1.2.1 公共参数设置.....1

1.2.2 公共返回值解析.....3

1.3 MQ Open API公共术语.....4

1.3.1 网关参数类.....4

1.3.1.1 RegionId.....4

1.3.1.2 EndPoint.....4

1.3.1.3 Domain.....4

1.3.2 账号安全类.....4

1.3.2.1 AccessKey.....4

1.3.2.2 SecretKey.....4

1.3.3 MQ资源管控类.....4

1.3.3.1 Topic.....4

1.3.3.2 Producer ID（PID）.....4

1.3.3.3 Consumer ID（CID）.....4

1.3.3.4 消息堆积.....5

1.4 MQ Open API错误码列表.....5

1.5 API详细说明.....7

1.5.1 Region管理.....7

1.5.1.1 获取regionId列表（/json/region/list）.....7

1.5.2 Topic管理.....8

1.5.2.1 创建Topic（/json/topic/create）.....8

1.5.2.2 删除Topic（/json/topic/delete）.....9

1.5.2.3 查询Topic（/json/topic/get）.....10

1.5.2.4 查询Topic名字（/json/topic/listNames）.....10

1.5.2.5 查询Topic列表（/json/topic/listInPage）.....10

1.5.2.6 查询Topic状态（/json/topic/status）.....11

1.5.3 发布管理.....11

1.5.3.1 发布信息注册（/json/publish/create）.....11

1.5.3.2 发布信息删除（/json/publish/delete）.....12

1.5.3.3 发布信息获取（/json/publish/get）.....12

1.5.3.4 获取发布列表（/json/publish/list）.....13
1.5.4 订阅管理.....13
1.5.4.1 注册订阅信息（/json/subscription/create）.....13
1.5.4.2 删除订阅信息（/json/subscription/delete）.....14
1.5.4.3 获取订阅信息（/json/subscription/get）.....14
1.5.4.4 获取订阅列表（/json/subscription/list）.....15
1.5.5 消息查询.....15
1.5.5.1 按Message ID查询消息（/json/message/getById）.....15
1.5.5.2 按Message Key查询消息（/json/message/getByKey）.....16
1.5.5.3 按Topic查询消息（/json/message/getByTopic）.....16
1.5.6 消费管理.....17
1.5.6.1 消费堆积查询（/json/consumer/accumulate）.....17
1.5.6.2 消费状态查询（/json/consumer/status）.....17
1.5.6.3 消费者连接查询（/json/consumer/getConnection）.....18
1.5.6.4 消费进度查询（/json/consumer/progress）.....18
1.5.7 授权管理.....19
1.5.7.1 创建主账号授权（/json/empower/create）.....19
1.5.7.2 创建子账号授权（/json/empower/createBySub）.....19
1.5.7.3 删除授权（/json/empower/delete）.....20
1.5.7.4 查询授权列表（/json/empower/list）.....21
1.5.8 MQTT管理（如果有部署）.....21
1.5.8.1 查询分组列表（/json/mqtt/listGroupId）.....21
1.5.8.2 查询设备在线信息（/json/mqtt/getDeviceRunningInfo）.....21
1.5.8.3 查询分组在线信息（/json/mqtt/getClientSetByGroupId）.....22
1.5.8.4 根据Topic查询在线信息（/json/mqtt/getClientSetByTopic）.....22
1.5.9 实例管理.....23
1.5.9.1 实例网络类型切换（/json/vpcTunnel/tunnelSwitch）.....23
1.5.9.2 获取实例列表（/json/instance/list）.....24
1.5.9.3 获取实例ID列表（/json/instance/listIds）.....24
2 SDK参考.....26
2.1 概述.....26
2.2 消息模型.....26
2.3 传输协议与字符编码.....26
2.4 名词解释.....27
2.5 传输接口规范.....28
2.5.1 常量类PropertyKeyConst.....28
2.5.2 工厂类ONSFactory.....29
2.5.2.1 createConsumer.....29
2.5.2.2 createBatchConsumer.....29
2.5.2.3 createProducer.....30
2.5.3 消息类Message.....30

2.5.3.1 Message默认构造方法.....30
2.5.3.2 Message有参构造方法.....30
2.5.3.3 Message有参构造方法.....31
2.5.3.4 putUserProperties.....31
2.5.3.5 getUserProperties.....32
2.5.3.6 getTopic.....32
2.5.3.7 setTopic.....32
2.5.3.8 getTag.....32
2.5.3.9 setTag.....33
2.5.3.10 getKey.....33
2.5.3.11 setKey.....33
2.5.3.12 getMsgID.....33
2.5.3.13 setMsgID.....34
2.5.3.14 getReconsumeTimes.....34
2.5.3.15 setReconsumeTimes.....34
2.5.3.16 getBornTimestamp.....35
2.5.3.17 setBornTimestamp.....35
2.5.3.18 getBornHost.....35
2.5.3.19 setBornHost.....35
2.5.3.20 setStartDeliverTime.....36
2.5.3.21 getStartDeliverTime.....36
2.5.3.22 setShardingKey.....36
2.5.3.23 getShardingKey.....36
2.5.4 异常接口.....37
2.5.4.1 ONSClientException 默认构造方法.....37
2.5.4.2 ONSClientException 有参构造方法.....37
2.5.4.3 ONSClientException 有参构造方法.....37
2.5.4.4 ONSClientException 有参构造方法.....37
2.5.5 发送消息接口说明.....38
2.5.5.1 发送结果SendResult.....38
2.5.5.1.1 SendResult 默认构造方法.....38
2.5.5.1.2 getMessageId.....38
2.5.5.1.3 getTopic.....39
2.5.5.1.4 toString.....39
2.5.5.2 生产者实例 Producer.....39
2.5.5.2.1 send.....39
2.5.5.2.2 sendOneway.....40
2.5.5.2.3 sendAsync.....40
2.5.5.2.4 isStarted.....40
2.5.5.2.5 isClosed.....41
2.5.6 订阅消息接口说明.....41
2.5.6.1 消费者实例 Consumer.....41
2.5.6.1.1 subscribe.....41

2.5.6.1.2 unsubscribe....42
2.5.6.1.3 isStarted....42
2.5.6.1.4 isClosed....42
2.5.6.1.5 start....43
2.5.6.1.6 shutdown....43
2.5.6.2 消息监听器 MessageListener....43
2.5.6.2.1 consume....43
2.5.6.3 消息消费结果 Action....44
2.5.6.3.1 valueOf....44
2.5.6.3.2 values....44
2.5.6.4 消费模式 PropertyValueConst....45
2.5.6.5 批量消费者实例 BatchConsumer....45
2.5.6.5.1 subscribe....45
2.5.6.5.2 unsubscribe....45
2.5.6.5.3 isStarted....46
2.5.6.5.4 isClosed....46
2.5.6.5.5 start....46
2.5.6.5.6 shutdown....46
2.5.6.6 批量消费消息监听器 BatchMessageListener....47
2.5.6.6.1 consume....47

1 API参考

1.1 MQ Open API简介

本文档介绍MQ Open API的基本概念，包括使用场景、基本原理、功能以及局限性。

1.1.1 面向用户和场景

Open API是MQ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MQ相关的资源，例如Topic、Consumer ID等。

Open API的主要用户：

- 已经熟悉MQ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。

1.1.2 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MQ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。

1.1.3 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MQ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。

1.2 MQ Open API接入指南

本章节介绍MQ Open API的接入步骤以及注意事项，包含公共参数设置，以及公共返回值的解析。

1.2.1 公共参数设置

客户端基于HTTP方式调用MQ 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	当前时间戳（单位ms），判断请求是否失效
_platform	String	是	调用类型，固定值：onsConsole
_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，专有域和私有云情况下，默认值为mq-private。

_signature参数是对HTTP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：

```
http://ons.console.example.com/json/topic/create
```

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。

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：

```
http://ons.console.example.com/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作为秘钥，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+adwe23fssf**

最后，整个HTTP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）：

```
http://ons.console.example.com/json/topic/create?Key_2=Value_2&Key_1=Value_1&Key_3=
Value_3&_signature= eqweq%2Badwe23fssf%3D
```

1.2.2 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": "",
   "helpUrl": null,
   "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。

1.3 MQ Open API公共术语

本章节给出MQ Open API涉及到的公共术语的解释，供查询参考。

1.3.1 网关参数类

1.3.1.1 RegionId

Region信息，使用不同的regionId进行环境隔离。调用Open API时需要指定操作哪个regionId。详细信息请参考*Region*管理。

1.3.1.2 EndPoint

Open API根据设置的接入点来调用对应的后端服务。一般设置为对应的RegionId即可。

1.3.1.3 Domain

Open API根据设置的访问域名来调用对应的后端服务。

1.3.2 账号安全类

1.3.2.1 AccessKey

访问阿里云产品使用的账号的AK，由阿里云官网获得。调用时传入给MQ后台用于身份验证，调用MQ Open API只能操作对应账号下的资源。

1.3.2.2 SecretKey

访问阿里云产品使用的账号的AK秘钥，由阿里云官网获得，调用时传入，用于计算访问请求的签名。MQ后端根据对应账号比对签名实现鉴权。

1.3.3 MQ资源管控类

1.3.3.1 Topic

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

1.3.3.2 Producer ID（PID）

MQ产品中确定发布关系的唯一标识，用于标识一个发送的集群。MQ中Producer ID全局唯一，后续文档中对Producer ID简称PID。

1.3.3.3 Consumer ID（CID）

MQ产品中确定订阅关系的唯一标识，用于标识一个订阅组。采用同一个Consumer ID的客户端，其消费消息的逻辑必须完全一致。MQ中Consumer ID全局唯一，后续文档中对Consumer ID简称CID。

1.3.3.4 消息堆积

消息堆积指MQ产品中描述消费方尚未进行处理的消息数据。当消息的生产快于消费的时候，两者之差即为堆积数，正常情况下堆积数应该在可控范围内。如果堆积数过多而且成上涨趋势，则需要关注消费方慢的原因。

1.4 MQ Open API错误码列表

MQ 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	MQ后端异常	通过阿里云工单联系MQ技术人员
ONS_SERVICE_UNSUPPORTED	当前调用在对应的Region区域不支持	请找MQ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MQ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CID找不到	请检查CID是否创建过，或者查询条件错误
BIZ_PUBLISHER_EXISTED	指定PID已经存在	更换PID的名称重试请求
BIZ_SUBSCRIPTION_EXISTED	指定CID已经存在	更换CID的名称重试请求
BIZ_CONSUMER_NOT_ONLINE	指定CID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

错误码	现象&原因	解决方法
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PID已经被其他用户占用	更换PID重试请求
BIZ_CONSUMER_ID_BELONG_TO_OTHER_USER	当前CID已经被其他用户占用	更换CID重试请求
BIZ_PUBLISH_INFO_NOT_FOUND	请求的PID没有找到	确认PID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
ONS_SYSTEM_ERROR	MQ后端异常	通过阿里云工单联系MQ技术人员
ONS_SERVICE_UNSUPPORTED	当前调用在对应的Region区域不支持	请找MQ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MQ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CID找不到	请检查CID是否创建过，或者查询条件错误
BIZ_PUBLISHER_EXISTED	指定PID已经存在	更换PID的名称重试请求
BIZ_SUBSCRIPTION_EXISTED	指定CID已经存在	更换CID的名称重试请求

错误码	现象&原因	解决方法
BIZ_CONSUMER_NOT_ONLINE	指定CID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PID已经被其他用户占用	更换PID重试请求
BIZ_CONSUMER_ID_BELONG_TO_OTHER_USER	当前CID已经被其他用户占用	更换CID重试请求
BIZ_PUBLISH_INFO_NOT_FOUND	请求的PID没有找到	确认PID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

1.5 API详细说明

1.5.1 Region管理

1.5.1.1 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（排除_regionId之外的所有公共参数）

参数名称	类型	是否必须	描述
_accesskey	String	是	用户账号AK
_preventCache	Long	是	当前时间戳（单位ms），判断请求是否失效
_platform	String	是	调用类型，固定值：onsConsole
_signature	String	是	请求的签名

返回参数列表

返回结果中data为空。

1.5.2 Topic管理

1.5.2.1 创建Topic（/json/topic/create）

创建Topic接口，该接口用于创建指定Topic到目标Region上。

使用场景

Topic创建接口一般是在有新的Topic需求的情况下调用，例如发布新应用，业务逻辑需要新的Topic。

Topic为全局唯一，不可重复创建，否则会创建失败，需要根据错误码进行处理。

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	需要创建的Topic名称，Topic名称全局唯一，不可重复
instanceId	String	是	创建Topic至指定实例集群

返回参数列表

返回结果中data为空。

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 'SECRETKET_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%2BrOBzOd0M2SNj3AcuqnqpSA%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}
```

1.5.2.2 删除Topic（/json/topic/delete）

此接口用于删除账号下的指定Topic。

使用场景

Topic删除接口一般是资源回收时使用，例如应用下线。Topic删除时MQ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。

说明：

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

返回参数列表

返回结果中data为空。

1.5.2.3 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

返回参数列表

以JSON的格式返回Topic的一些元数据信息。

1.5.2.4 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。

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。

1.5.2.5 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	否	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。

1.5.2.6 查询Topic状态（/json/topic/status）

调用该接口可以查询指定Topic当前的消息状态。

使用场景

用来查询Topic的队列，以及各队列的消息数量。

请求参数列表

其他参数请参考公共参数设置。

名称	类型	是否必须	描述
topic	String	是	指定查询的Topic名称

返回参数列表

以JSON的格式返回Topic的队列信息。

1.5.3 发布管理

1.5.3.1 发布信息注册（/json/publish/create）

此接口用于向MQ申请注册发布关系，并得到生成的PID，用于生产环境发消息。

使用场景

在创建Topic之后，需要注册对应的发布者，即生产者（PID）来发消息。

说明：

每个Topic只需要一个PID即可，不需要创建多个PID，因为PID可以在多个应用中复用。

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
ProducerId	String	是	用于消息发布的PID
topic	String	是	发布的Topic名称，必须已通过创建*Topic#/json/topic/create#*创建

返回参数列表

返回结果中data为空。

1.5.3.2 发布信息删除（/json/publish/delete）

调用该接口可以删除已经发布过的PID。

使用场景

应用下线或者资源释放时，需要删除已经存在的发布关系，调用该接口删除，删除前提需要确保发布关系已经存在。

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
producerId	String	否	需要删除的发布关系的PID
topic	String	是	需要删除的发布关系对应的Topic

返回参数列表

返回结果中data为空。

1.5.3.3 发布信息获取（/json/publish/get）

调用该接口可以查询已经发布过的PID。

使用场景

获取已创建的发布信息。

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
producerId	String	是	需要查询的发布关系的PID
topic	String	是	需要查询发布关系对应的Topic

返回参数列表

以JSON格式返回已创建的发布信息。

1.5.3.4 获取发布列表（/json/publish/list）

调用该接口查询已经发布过的PID列表。

使用场景

用于获取已创建的发布列表。

请求参数列表

其他参数请参考公共参数设置。

名称	类型	是否必须	描述
producerId	String	否	需要查询的发布关系的PID
topic	String	否	需要查询的发布关系对应的Topic
instanceId	String	否	查询指定实例下的发布列表

返回参数列表

以JSON格式返回已创建的发布列表。

1.5.4 订阅管理

1.5.4.1 注册订阅信息（/json/subscription/create）

调用该接口向MQ申请注册订阅关系，并得到生成的CID，用于生产环境收消息。

使用场景

新应用上线时，在注册Topic资源后，如果需要订阅消息，需要首先在MQ控制台或者调用该API申请注册对目标Topic的注册关系，然后才能使用注册的CID订阅消息。

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
consumerId	String	是	用于订阅的CID
topic	String	是	订阅的Topic

返回参数列表

返回结果中data为空。

1.5.4.2 删除订阅信息（/json/subscription/delete）

调用该接口可以删除之前使用/json/subscription/create接口创建的订阅关系。

使用场景

应用下线时销毁MQ的资源可以调用该接口删除订阅关系信息。

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
consumerId	String	是	需要删除的CID
topic	String	是	需要查询的Topic

返回参数列表

返回结果中data为空。

1.5.4.3 获取订阅信息（/json/subscription/get）

此接口用于获取指定Topic和CID订阅信息。

使用场景

获取已创建的CID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
consumerId	String	是	需要查询的CID
topic	String	是	需要查询的Topic

返回参数列表

以JSON格式返回已创建的CID信息。

1.5.4.4 获取订阅列表（/json/subscription/list）

此接口用于获取指定Topic和CID订阅列表。

使用场景

获取已创建的CID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	是否必须	描述
consumerId	String	否	需要查询的CID
topic	String	否	需要查询的Topic
instanceId	String	否	查询指定实例下的订阅列表

返回参数列表

以JSON格式返回已创建的CID列表。

1.5.5 消息查询

1.5.5.1 按Message ID查询消息（/json/message/getById）

此接口通过消息的消息ID查询已成功发送了的消息。

使用场景

根据消息ID查询已经发送成功的消息。

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
msgId	String	是	需要查询的消息的MsgId
topic	String	是	需要查询的Topic

返回参数列表

以JSON的方式返回消息的元数据信息。

1.5.5.2 按Message Key查询消息（/json/message/getByKey）

此接口通过消息的key查询已成功发送了的消息。

使用场景

根据MsgKey查询已发送成功的消息。

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
topic	String	是	需要查询的Topic
key	String	是	需要查询的消息的Key

返回参数列表

以JSON方式返回消息的元数据信息。

1.5.5.3 按Topic查询消息（/json/message/getByTopic）

此接口通过消息的Topic查询已成功发送了的消息。

使用场景

根据消息Topic查询已成功发送了的消息，消息量比较大，不建议调用此接口。

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	需要查询的消息所属的Topic

返回参数列表

以JSON方式返回消息的元数据信息。

1.5.6 消费管理

1.5.6.1 消费堆积查询（/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
consumerId	String	是	需要查询的消费端CID
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

以JSON的格式返回对应ConsumerId的堆积情况，如果带上detail=true的参数将返回各个队列中消息的堆积的详情。

1.5.6.2 消费状态查询（/json/consumer/status）

此接口用于查询指定订阅组的消费状态，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

消费者状态查询接口用于定位订阅组的详细状态，一般用于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。可以判断出指定消费组（CID）的订阅关系是否一致，负载均衡是否正常，以及在线客户端的Jstack信息等。

说明：
消费者状态查询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
consumerId	String	是	需要查询的消费端CID
detail	Boolean	否（默认false）	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
needJstack	Boolean	否（默认false）	是否返回客户端的Jstack信息

返回参数列表

以JSON的格式返回对应ConsumerId的消费状态。

1.5.6.3 消费者连接查询（/json/consumer/getConnection）

该接口用于查询指定订阅组(CID)下当前客户端的连接情况。

使用场景

一般用于判断指定CID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	是否必须	描述
consumerId	String	是	需要查询的消费端CID

返回参数列表

以JSON格式返回对应ConsumerId的连接列表。

1.5.6.4 消费进度查询（/json/consumer/progress）

消费进度查询接口，用于查询指定订阅组(CID)当前的消费进度信息。

使用场景

一般用于判断指定CID的消费者消费各个分区消息的进度。

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
consumerId	String	是	消费者的CID
topic	String	是	需要查询的Topic

返回参数列表

以JSON的格式返回对应ConsumerId消费该Topic的各个分区的位点信息。

1.5.7 授权管理

1.5.7.1 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

MQ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。

授权注意事项：

- 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CID，而不是使用Owner账号的CID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
empowerUser	Long	是	授权的目标主账号的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，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。

1.5.7.2 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

MQ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。

授权注意事项：

- 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CID，而不是使用Owner账号的CID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
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，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。

1.5.7.3 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。

注意事项：

- 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
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

返回data为null。

1.5.7.4 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

名称	类型	是否必须	描述
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。

1.5.8 MQTT管理（如果有部署）

1.5.8.1 查询分组列表（/json/mqtt/listGroupId）

该接口用于查询已经申请的 MQTT 分组列表。

使用场景

获取已创建的GroupId列表。

请求参数列表

请参考公共参数设置。

返回参数列表

以JSON格式返回已创建的GroupId列表。

1.5.8.2 查询设备在线信息（/json/mqtt/getDeviceRunningInfo）

查询单一设备在线信息的接口，调用该接口可以查询指定的MQTT客户端的状态信息。

使用场景

查询单个MQTT客户端的订阅关系和在线状态。

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
clientId	String	是	需要查询的 MQTT客户端的ID

返回参数

以JSON格式返回该设备的订阅关系以及在线状态。

1.5.8.3 查询分组在线信息（/json/mqtt/getClientSetByGroupId）

查询某个设备分组当前在线终端数量的接口，调用该接口可以查询指定的GroupId 下面在线客户端的数量信息。

使用场景

统计指定的MQTT GroupId下有多少设备在线。

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
groupId	String	是	需要查询的 MQTT分组ID

返回参数

以JSON格式返回该分组下在线终端的数量。

1.5.8.4 根据Topic查询在线信息（/json/mqtt/getClientSetByTopic）

根据Topic查询在线终端数量的接口，调用该接口可以查询订阅指定Topic的在线客户端的数量信息。

使用场景

统计业务层面订阅某个Topic的设备在线数。

请求参数列表

其他参数请参考公共参数设置。

名称	类型	是否必须	描述
firstTopic	String	是	需要查询的 一级Topic
secondTopic	String	否	需要查询的子级Topic，如果没有则不填

返回参数

以JSON格式返回订阅该Topic的在线终端的数量。

1.5.9 实例管理

1.5.9.1 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
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

名称	类型	是否必须	描述
			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。

1.5.9.2 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

其他参数请参考公共参数设置。

返回参数列表

以JSON格式返回实例列表。

1.5.9.3 获取实例ID列表（/json/instance/listIds）

获取当前环境所有实例ID的列表。

使用场景

查询所有的实例ID，供搜索条件输入或创建Topic时输入。

请求参数列表

其他参数请参考公共参数设置。

返回参数列表

以JSON格式返回实例ID列表。

2 SDK参考

2.1 概述

MQ 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。

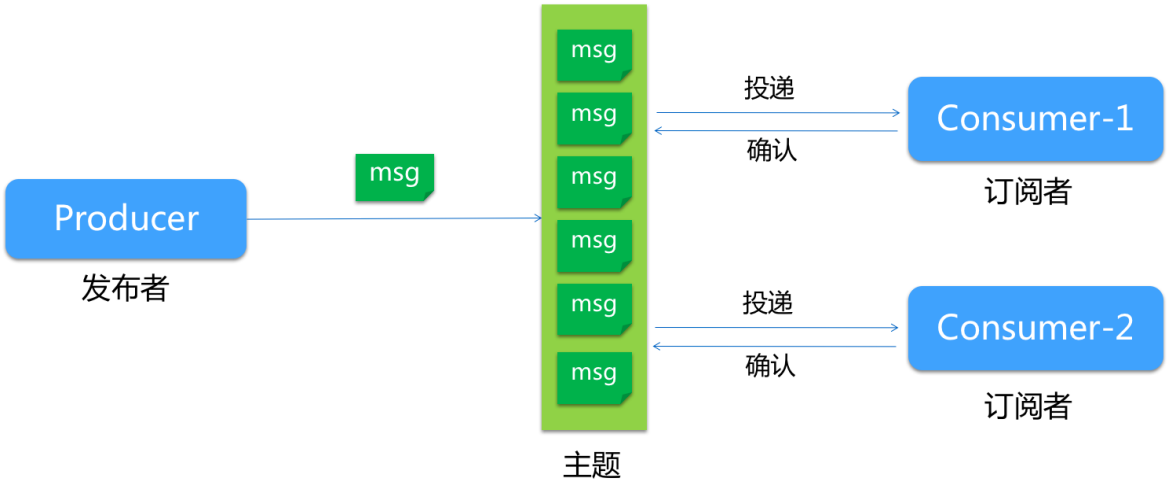
本标准规定消息队列（Message Queue，简称 MQ）应用节点与消息队列服务中心间数据传输服务管理方法，包括服务遵循的规范，以及服务管理相关的公共API。

本标准适用于 Java 语言 SDK，以 TCP 协议 MQ 进行消息传输方式。

2.2 消息模型

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

图 2-1: Pub-Sub模型



2.3 传输协议与字符编码

MQ 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

MQ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。

2.4 名词解释

Topic	消息主题，一级消息类型，通过 Topic 对消息进行分类。
Message	消息，消息队列中信息传递的载体。
Message ID	消息的全局唯一标识，由 MQ 系统自动生成，唯一标识某条消息。
Message Key	消息的业务标识，由消息生产者（Producer）设置，唯一标识某个业务逻辑。
Tag	消息标签，二级消息类型，用来进一步区分某个 Topic 下的消息分类。
Producer ID	一类 Producer 的标识，这类 Producer 通常生产并发送一类消息，且发送逻辑一致。
Producer 实例	Producer 的一个对象实例，不同的 Producer 实例可以运行在不同进程内或者不同机器上。Producer 实例线程安全，可在同一进程内多线程之间共享。
Consumer ID	一类 Consumer 的标识，这类 Consumer 通常接收并消费一类消息，且消费逻辑一致。
Consumer 实例	Consumer 的一个对象实例，不同的 Consumer 实例可以运行在不同进程内或者不同机器上。一个 Consumer 实例内配置线程池消费消息。
集群消费	一个 Consumer ID 所标识的所有 Consumer 平均分摊消费消息。例如某个 Topic 有 9 条消息，一个 Consumer ID 有 3 个 Consumer 实例，那么在集群消费模式下每个实例平均分摊，只消费其中的 3 条消息。
广播消费	一个 Consumer ID 所标识的所有 Consumer 都会各自消费某条消息一次。例如某个 Topic 有 9 条消息，一个 Consumer ID 有 3 个 Consumer 实例，那么在广播消费模式下每个实例都会各自消费 9 条消息。
消息过滤	订阅者可以根据消息标签（Tag）对消息进行过滤，确保订阅者最终只接收被过滤后的消息类型。消息过滤在 MQ 服务端完成。
定时消息	Producer 将消息发送到 MQ 服务端，但并不期望这条消息立马投递，而是推迟到在当前时间点之后的某一个时间投递到 Consumer 进行消费，该消息即定时消息。
延时消息	Producer 将消息发送到 MQ 服务端，但并不期望这条消息立马投递，而是延迟一定时间后才投递到 Consumer 进行消费，该消息即延时消息。

2.5 传输接口规范

本单元主要说明 MQ SDK 公共接口、常量以及常用对象。

2.5.1 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，用于身份认证。
SecretKey	String	-	您在阿里云账号管理控制台中创建的SecretKey，用于身份认证。
OnsChannel	String	ALIYUN	用户渠道，阿里云：ALIYUN，聚石塔用户为：CLOUD。
ONSAddr	String	-	设置 MQ TCP 协议接入点。
NAMESRV_ADDR	String	-	设置 Name Server 列表（不推荐），与 ONSAddr 二选一。
ProducerId	String	-	Producer 实例的唯一ID，您在控制台创建的 Producer ID。
ConsumerId	String	-	Consumer 实例的唯一ID，您在控制台创建的 Consumer 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。

2.5.2 工厂类ONSFactory

MQ SDK工厂类，用来创建客户端发布者、订阅者实例，其所有的方法都包含一个Properties对象，用来传递必要的参数。

2.5.2.1 createConsumer

- 语法：static Consumer createConsumer(Properties properties)
- 描述：创建 MQ 普通消息、定时/延时消息等消息类型的消费者实例。
- 参数：

名称	类型	描述
properties	Properties	详情参考 PropertyKeyConst 说明。 必填属性：PropertyKeyConst.ONSAddr、PropertyKeyConst.ConsumerId、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey； 可选属性：PropertyKeyConst.ConsumeThreadNums、 PropertyKeyConst.ConsumeTimeout。

- 返回值

类型	描述
Consumer	MQ 普通消息、定时/延时消息等消息类型的消费者实例

2.5.2.2 createBatchConsumer

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

名称	类型	描述
properties	Properties	详情参考 PropertyKeyConst 说明; 必填属性：PropertyKeyConst.ONSAddr、PropertyKeyConst.ConsumerId、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey； 可选属性：PropertyKeyConst.ConsumeThreadNums、PropertyKeyConst.ConsumeTimeout、PropertyKeyConst.ConsumeMessageBatchMaxSize。

- 返回值：

类型	描述
BatchConsumer	MQ 批量消费者实例

2.5.2.3 createProducer

- 语法：static Producer createProducer(Properties properties)
- 描述：创建 MQ 普通消息、定时/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.ONSAddr、PropertyKeyConst.ProducerId、PropertyKeyConst.AccessKey、PropertyKeyConst.SecretKey； 可选属性： PropertyKeyConst.NAMESRV_ADDR、PropertyKeyConst.SendMsgTimeoutMillis。

- 返回值：

类型	描述
Producer	MQ 普通消息、定时/延时消息发布者实例

2.5.3 消息类Message

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。

2.5.3.1 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

2.5.3.2 Message有参构造方法

- 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4M，具体请参阅集群部署文档描述。

2.5.3.3 Message有参构造方法

- 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4M，具体请参阅集群部署文档描述。

2.5.3.4 putUserProperties

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

2.5.3.5 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

2.5.3.6 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

2.5.3.7 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

2.5.3.8 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

2.5.3.9 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

2.5.3.10 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

2.5.3.11 setKey

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

名称	类型	描述
key	String	业务标识

- 返回值：无

2.5.3.12 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

2.5.3.13 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

2.5.3.14 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

2.5.3.15 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

2.5.3.16 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

2.5.3.17 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

2.5.3.18 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

2.5.3.19 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

2.5.3.20 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

2.5.3.21 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

2.5.3.22 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

2.5.3.23 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

2.5.4 异常接口

统一对外暴露的客户端异常接口。

2.5.4.1 ONSClientException 默认构造方法

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

2.5.4.2 ONSClientException 有参构造方法

- 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

2.5.4.3 ONSClientException 有参构造方法

- 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

2.5.4.4 ONSClientException 有参构造方法

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

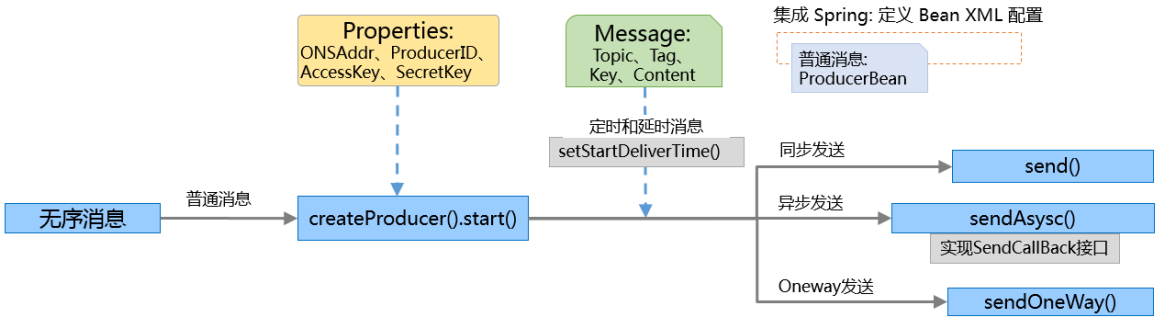
名称	类型	描述
message	String	需要向外传递的异常信息

名称	类型	描述
cause	Throwable	需要向外传递的异常

2.5.5 发送消息接口说明

本单元主要详述MQ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。

图 2-2: 发布消息相关接口参数说明



2.5.5.1 发送结果SendResult

消息发送结果对象。

2.5.5.1.1 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

2.5.5.1.2 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

2.5.5.1.3 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

2.5.5.1.4 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

2.5.5.2 生产者实例 Producer

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

MQ 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。

2.5.5.2.1 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

2.5.5.2.2 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

2.5.5.2.3 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

2.5.5.2.4 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

2.5.5.2.5 isClosed

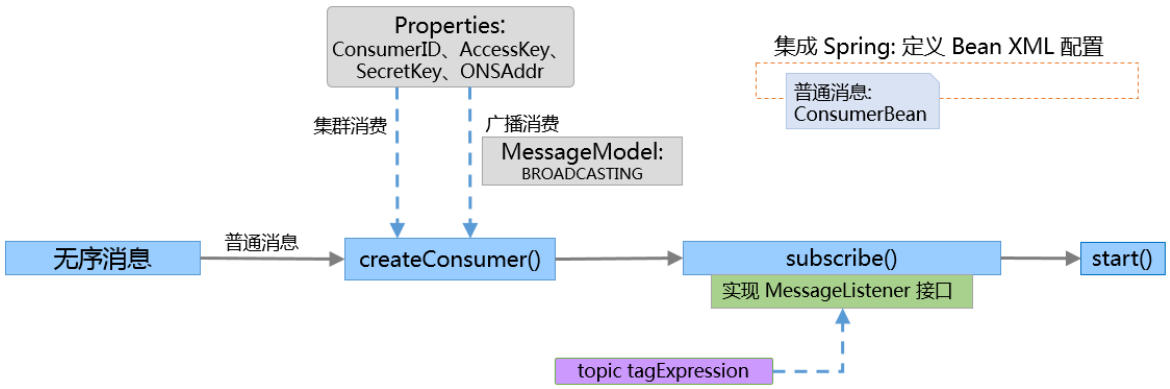
- 语法：public boolean isClosed()
- 描述：检查服务是否已经关闭。
- 参数：无
- 返回值：

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

2.5.6 订阅消息接口说明

本单元主要详述 MQ 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。

图 2-3: 订阅消息接口相关参数说明



2.5.6.1 消费者实例 Consumer

消息消费者实例，用来订阅消息，支持普通消息、定时消息/延时消息。

2.5.6.1.1 subscribe

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

名称	类型	描述
topic	String	消息主题

名称	类型	描述
subExpression	String	订阅过滤表达式字符串，MQ 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	MessageListener	消息回调监听器

- 返回值：无

2.5.6.1.2 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

2.5.6.1.3 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

2.5.6.1.4 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

2.5.6.1.5 start

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

2.5.6.1.6 shutdown

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

2.5.6.2 消息监听器 MessageListener

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

2.5.6.2.1 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。</br> 消费失败：如果应用抛出异常或者返回Null等价于返回 Action.ReconsumeLater；</br> 消费成功：返回 Action.CommitMessage；

2.5.6.3 消息消费结果 Action

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|

2.5.6.3.1 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

2.5.6.3.2 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

2.5.6.4 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

2.5.6.5 批量消费者实例 BatchConsumer

批量消息消费者，用来通过批量的方式订阅消息。

2.5.6.5.1 subscribe

- 语法：void subscribe(String topic, String subExpression, BatchMessageListener listener)
- 描述：订阅消息，包括普通消息、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，MQ 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

2.5.6.5.2 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

2.5.6.5.3 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

2.5.6.5.4 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

2.5.6.5.5 start

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

2.5.6.5.6 shutdown

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

2.5.6.6 批量消费消息监听器 BatchMessageListener

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。

2.5.6.6.1 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；