

Alibaba Cloud

消息队列 MQ
OpenAPI 参考

文档版本：20201028

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.调用方式	07
2.服务接入点	08
3.获取 SDK	10
4.API 概览	11
5.API 版本说明	14
6.签名机制	15
7.公共参数	18
8.获取 AccessKey	21
9.错误码列表	22
10.服务开通接口	23
10.1. OpenOnsService	23
11.Region 管理接口	25
11.1. OnsRegionList	25
12.实例管理接口	31
12.1. OnsInstanceCreate	31
12.2. OnsInstanceBaseInfo	32
12.3. OnsInstanceDelete	36
12.4. OnsInstanceUpdate	37
12.5. OnsInstanceInServiceList	39
13.Topic 管理接口	43
13.1. OnsTopicCreate	43
13.2. OnsTopicDelete	44
13.3. OnsTopicStatus	46
13.4. OnsTopicList	47
13.5. OnsTopicUpdate	51
13.6. OnsTopicSubDetail	53

14.Group 管理接口	56
14.1. OnsGroupCreate	56
14.2. OnsGroupDelete	57
14.3. OnsGroupList	58
14.4. OnsGroupSubDetail	62
14.5. OnsGroupConsumerUpdate	65
15.标签管理接口	67
15.1. TagResources	67
15.2. ListTagResources	69
15.3. UntagResources	71
16.消费管理接口	74
16.1. OnsConsumerAccumulate	74
16.2. OnsConsumerStatus	77
16.3. OnsConsumerGetConnection	86
16.4. OnsConsumerResetOffset	88
16.5. OnsConsumerTimeSpan	90
16.6. OnsMessagePush	92
17.消息查询接口	95
17.1. OnsMessageTrace	95
17.2. OnsMessageGetByMsgId	97
17.3. OnsMessageGetByKey	101
17.4. OnsMessagePageQueryByTopic	105
18.发布订阅统计接口	111
18.1. OnsTrendTopicInputTps	111
18.2. OnsTrendGroupOutputTps	117
19.消息轨迹接口	124
19.1. OnsTraceQueryByMsgId	124
19.2. OnsTraceGetResult	125

19.3. OnsTraceQueryByMsgKey	131
20.死信队列接口	133
20.1. OnsDLQMessageGetById	133
20.2. OnsDLQMessagePageQueryByGroupId	137
20.3. OnsDLQMessageResendById	142

1.调用方式

消息队列RocketMQ版

产品接口支持 HTTP 调用、SDK 调用和 OpenAPI Explorer 调用。

HTTP 调用

消息队列RocketMQ版

的 API 是 RPC 风格，您可以通过发送 HTTP GET 请求调用

消息队列RocketMQ版

的 API。

其请求结构如下：

```
http://Endpoint/?Action=xx&Parameters
```

其中：

- Endpoint：
消息队列RocketMQ版
的 API 的服务接入地址为 `ons.[regionId].aliyuncs.com`。[regionId] 详情请参见[服务接入点](#)。
- Action：要执行的操作，如调用 ApplyToken 来申请 Token。
- Version：要使用的 API 版本，
消息队列RocketMQ版
的 API 版本是 `2019-02-14`。
- Parameters：请求参数，每个参数之间用 “&” 分隔。
请求参数由公共请求参数和 API 自定义参数组成。公共参数中包含 API 版本号、身份验证等信息，详情请参见[公共参数](#)。

SDK 调用

阿里云

消息队列RocketMQ版

提供多语言的 SDK。阿里云 SDK 免去您手动签名的过程，方便使用。详情请参见[获取 SDK](#)。

OpenAPI Explorer 调用

OpenAPI Explorer 可视化的 API 调用工具。通过该工具，您可以通过网页或者命令行调用各云产品以及 API 市场上开放的 API，查看每次的 API 请求和返回结果，并生成相应 SDK 调用示例。

您可以直接访问 [OpenAPI Explorer](#) 调用 API 也可以通过 API 文档中的调试功能进行调用。

2. 服务接入点

本文列举

消息队列RocketMQ版

的 API 接入地域 ID 和域名地址。

地域名称	regionId	domain
公网	mq-internet-access	ons.mq-internet-access.aliyuncs.com
华东1（杭州）	cn-hangzhou	ons.cn-hangzhou.aliyuncs.com
华东2（上海）	cn-shanghai	ons.cn-shanghai.aliyuncs.com
华北1（青岛）	cn-qingdao	ons.cn-qingdao.aliyuncs.com
华北2（北京）	cn-beijing	ons.cn-beijing.aliyuncs.com
华北3（张家口）	cn-zhangjiakou	ons.cn-zhangjiakou.aliyuncs.com
华北5（呼和浩特）	cn-huhehaote	ons.cn-huhehaote.aliyuncs.com
华南1（深圳）	cn-shenzhen	ons.cn-shenzhen.aliyuncs.com
西南1（成都）	cn-chengdu	ons.cn-chengdu.aliyuncs.com
中国香港	cn-hongkong	ons.cn-hongkong.aliyuncs.com
新加坡（新加坡）	ap-southeast-1	ons.ap-southeast-1.aliyuncs.com
澳大利亚（悉尼）	ap-southeast-2	ons.ap-southeast-2.aliyuncs.com
马来西亚（吉隆坡）	ap-southeast-3	ons.ap-southeast-3.aliyuncs.com
印度尼西亚（雅加达）	ap-southeast-5	ons.ap-southeast-5.aliyuncs.com
日本（东京）	ap-northeast-1	ons.ap-northeast-1.aliyuncs.com
德国（法兰克福）	eu-central-1	ons.eu-central-1.aliyuncs.com
英国（伦敦）	eu-west-1	ons.eu-west-1.aliyuncs.com
美国（弗吉尼亚）	us-east-1	ons.us-east-1.aliyuncs.com
美国（硅谷）	us-west-1	ons.us-west-1.aliyuncs.com
印度（孟买）	ap-south-1	ons.ap-south-1.aliyuncs.com
阿联酋（迪拜）	me-east-1	ons.me-east-1.aliyuncs.com

地域名称	regionId	domain
华东1 金融云	cn-hangzhou-finance	ons.cn-hangzhou-finance.aliyuncs.com
华南1 金融云	cn-shenzhen-finance-1	ons.cn-shenzhen-finance-1.aliyuncs.com
华东2 金融云	cn-shanghai-finance-1	ons.cn-shanghai-finance-1.aliyuncs.com
华北2 阿里政务云1	cn-north-2-gov-1	ons.cn-north-2-gov-1.aliyuncs.com

3. 获取 SDK

本文介绍

消息队列RocketMQ版

的 API 的各语言 SDK 的下载地址。

SDK 下载

消息队列RocketMQ版

的 API 的 SDK 支持以下 6 种语言，各语言对应的 SDK 下载地址如下表所示。

语言	下载地址
Java	Alibaba Cloud RocketMQ SDK for Java
Python	Alibaba Cloud RocketMQ SDK for Python
PHP	Alibaba Cloud RocketMQ SDK for PHP
.NET	Alibaba Cloud RocketMQ SDK for .NET
Go	Alibaba Cloud RocketMQ SDK for Go
C++	Alibaba Cloud RocketMQ SDK for CPP

API 的 SDK 历史版本，请参见 [API 版本说明](#)。

4.API 概览

消息队列RocketMQ版

提供以下相关 API 接口。

地域管理接口

API	描述
OnsRegionList	使用 OnsRegionList 获取消息队列RocketMQ版目前开放服务的地域（Region）信息列表。

实例管理接口

API	描述
OnsInstanceBaseInfo	使用 OnsInstanceBaseInfo 查询实例基本信息和收发消息的接入点。
OnsInstanceCreate	使用 OnsInstanceCreate 创建实例。
OnsInstanceDelete	使用 OnsInstanceDelete 删除实例。
OnsInstanceInServiceList	使用 OnsInstanceInServiceList 查询当前用户账号下某地域所有实例的信息。
OnsInstanceUpdate	使用 OnsInstanceUpdate 更新实例名称和描述。

Topic 管理接口

API	描述
OnsTopicCreate	使用 OnsTopicCreate 创建 Topic。
OnsTopicDelete	使用 OnsTopicDelete 删除您账号下指定的 Topic。
OnsTopicList	使用 OnsTopicList 查询账号下所有 Topic 的信息列表。
OnsTopicStatus	使用 OnsTopicStatus 查询当前 Topic 下的消息总量以及 Topic 的最后更新时间。
OnsTopicUpdate	使用 OnsTopicUpdate 配置 Topic 的读写模式。
OnsTopicSubDetail	使用 OnsTopicSubDetail 查看有哪些在线订阅组订阅了这个 Topic。

Group 管理接口

API	描述
OnsGroupCreate	使用 OnsGroupCreate 创建客户端 Group ID。

API	描述
OnsGroupDelete	使用 OnsGroupDelete 删除之前使用 OnsGroupCreate 接口创建的 Group。
OnsGroupList	使用 OnsGroupList 展示您的 Group ID 资源的列表，并不用于具体的信息查询。
OnsGroupSubDetail	使用 OnsGroupSubDetail 查看 Group 订阅了哪些 Topic，如果 Group ID 对应的消费者实例不在线则查不到数据。
OnsGroupConsumerUpdate	使用 OnsGroupConsumerUpdate 配置指定 Group ID 对应的消费集群的消息读取权限。

标签管理接口

API	描述
TagResources	使用 TagResources 为资源绑定标签。
ListTagResources	使用 ListTagResources 查询资源绑定的标签列表。
UntagResources	使用 UntagResources 为资源解绑并删除标签。

消费管理接口

API	描述
OnsConsumerAccumulate	使用 OnsConsumerAccumulate 查询指定 Group ID 的消息消费堆积情况，包括当前消息堆积数量和消费延迟时间等。
OnsConsumerStatus	使用 OnsConsumerStatus 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。
OnsConsumerGetConnection	使用 OnsConsumerGetConnection 查询指定 Group ID 下当前客户端的连接情况。
OnsConsumerResetOffset	使用 OnsConsumerResetOffset 重置指定的 Group ID 的消费位点到指定时间戳。
OnsConsumerTimeSpan	使用 OnsConsumerTimeSpan 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。
OnsMessagePush	使用 OnsMessagePush 向指定的消费者推送消息。

消息查询接口

API	描述
OnsMessageTrace	使用 OnsMessageTrace 根据 Message ID 来判断目标消息是否曾被消费过。

API	描述
OnsMessageGetByMsgId	使用 OnsMessageGetByMsgId 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。
OnsMessageGetByKey	使用 OnsMessageGetByKey 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。
OnsMessagePageQueryByTopic	使用 OnsMessagePageQueryByTopic 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。

发布订阅统计接口

API	描述
OnsTrendTopicInputTps	使用 OnsTrendTopicInputTps 查询一段时间内指定的 Topic 的消息写入报表数据。
OnsTrendGroupOutputTps	使用 OnsTrendGroupOutputTps 查询指定的 Group ID 在一段时间内消费消息的统计信息。

消息轨迹接口

API	描述
OnsTraceGetResult	使用 OnsTraceGetResult 通过传入轨迹查询任务的 ID 获取之前的轨迹查询结果。
OnsTraceQueryByMsgId	使用 OnsTraceQueryByMsgId 通过传入 Topic 名称和 Message ID 创建轨迹查询任务，得到该查询任务的 ID。
OnsTraceQueryByMsgKey	使用 OnsTraceQueryByMsgKey 通过传入 Topic 名称和 Message Key 创建轨迹查询任务，得到该查询任务的 ID。

死信队列接口

API	描述
OnsDLQMessageGetById	使用 OnsDLQMessageGetById 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。
OnsDLQMessagePageQueryByGroupId	使用 OnsDLQMessagePageQueryByGroupId 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。
OnsDLQMessageResendById	使用 OnsDLQMessageResendById 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。

5.API 版本说明

本文介绍

消息队列RocketMQ版

的 API 版本发布信息。

详情请见下表。

API 版本	发布日期	说明
3.1.1	2019 年 3 月 28 日	- 新增查询在线订阅关系的接口 - 去除了部分接口的白名单限制，所有用户都可调用全部 API
3.1.0	2019 年 2 月 15 日	- 不兼容 2.x.x 版本，和线上控制台功能一致 - 提供实例化所有功能 - consumerId 和 producerId 概念变更为 groupId - 去除 OnsRegionId 参数，简化调用方式
2.1.4	2019 年 1 月 25 日	- 兼容 2.x.x 版本 - OnsTopicList 返回实例化信息字段
2.0.1	2018 年 4 月 17 日	新增创建 MQTT GroupId 的接口
2.0.0	2017 年 9 月 19 日	- 新增消息轨迹查询接口 - 写操作接口新增访问权限控制
1.3.3	2017 年 8 月 10 日	- 新增 Topic 禁读禁写接口 - 新增 ConsumerId 禁读接口

6. 签名机制

为保证 API 的安全调用，在调用 API 时阿里云会对每个 API 请求通过签名（Signature）进行身份验证。无论使用 HTTP 还是 HTTPS 协议提交请求，都需要在请求中包含签名信息。

概述

RPC API 要按以下格式在 API 请求的 Query 中增加签名（Signature）。

```
https://Endpoint/?SignatureVersion=1.0&SignatureMethod=HMAC-SHA1&Signature=CT9X0VtwR86fNWSnsc6v8YGOJuE%3D&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
```

 **说明** 推荐您使用阿里云 SDK 以免去手动签名的过程，方便使用。详情请参见[获取 SDK](#)。

- Endpoint：
消息队列RocketMQ版的 API 的服务接入地址为 `ons.[regionId].aliyuncs.com`。[regionId] 详情请参见[服务接入点](#)。
- SignatureMethod：签名方式，目前支持 HMAC-SHA1。
- SignatureVersion：签名算法版本，目前版本是 1.0。
- SignatureNonce：唯一随机数，用于防止网络重放攻击。您在不同请求间要使用不同的随机数值，建议使用通用唯一识别码 UUID（Universally Unique Identifier）。
- Signature：使用 AccessKey Secret 对请求进行对称加密后生成的签名。

签名算法遵循 RFC 2104 HMAC-SHA1 规范，使用 AccessSecret 对编码、排序后的整个请求串计算 HMAC 值作为签名。签名的元素是请求自身的一些参数，由于每个 API 请求内容不同，所以签名的结果也不尽相同。可参考本文的操作步骤，计算签名值。

```
Signature = Base64( HMAC-SHA1( AccessSecret, UTF-8-Encoding-Of( StringToSign) ) )
```

步骤一：构造待签名字符串

- 使用请求参数构造规范化的请求字符串（Canonicalized Query String）。
 - 按照参数名称的字典顺序对请求中所有的请求参数（包括公共请求参数和接口的自定义参数，但不包括公共请求参数中的 Signature 参数）进行排序。

 **说明** 当使用 GET 方法提交请求时，这些参数就是请求 URI 中的参数部分，即 URI 中“?”之后由“&”连接的部分。

ii. 对排序之后的请求参数的名称和值分别用 UTF-8 字符集进行 URL 编码。编码规则请参考下表。

字符	编码方式
A-Z、a-z 和 0-9 以及 “-”、“_”、“.” 和 “~”	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 进制表示。例如英文的双引号 (") 对应的编码为 %22。
扩展的 UTF-8 字符	编码成 %XY%ZA... 的格式。
英文空格	编码成 %20，而不是加号 (+)。 该编码方式和一般采用的 application/x-www-form-urlencoded MIME 格式编码算法（例如 Java 标准库中的 java.net.URLEncoder 的实现）存在区别。编码时可以先用标准库的方式进行编码，然后把编码后的字符串中的加号 (+) 替换成 %20，星号 (*) 替换成 %2A，%7E 替换回波浪号 (~)，即可得到上述规则描述的编码字符串。本算法可以用下面的 percentEncode 方法来实现： <pre>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; }</pre>

iii. 将编码后的参数名称和值用英文等号 (=) 进行连接。

iv. 将等号连接得到的参数组合按步骤 i 排好的顺序依次使用 “&” 符号连接，即得到规范化请求字符串。

2. 将第一步构造的规范化字符串按照下面的规则构造待签名的字符串。

```
StringToSign=
    HTTPMethod + "&" +
    percentEncode( "/" ) + "&" +
    percentEncode(CanonicalizedQueryString)
```

其中：

- HTTPMethod 是提交请求用的 HTTP 方法，例如 GET。
- percentEncode("/") 是按照上述的 URL 编码规则对字符 “/” 进行编码得到的值，即 %2F。
- percentEncode(CanonicalizedQueryString) 是对步骤 i 中构造的规范化请求字符串按步骤 ii 中描述的

URL 编码规则编码后得到的字符串。

步骤二：计算签名值

1. 按照 RFC2104 的定义，计算待签名字符串（StringToSign）的 HMAC 值。

❓ 说明 计算签名时使用的 Key 就是您持有的 AccessKey Secret 并加上一个 “&” 字符（ASCII:38），使用的哈希算法是 SHA1。

2. 按照 Base64 编码规则把上面的 HMAC 值编码成字符串，即得到签名值（Signature）。
3. 将得到的签名值作为 Signature 参数添加到请求参数中。

❓ 说明 得到的签名值在作为最后的请求参数值提交时要和其它参数一样，按照 RFC3986 的规则进行 URL 编码。

示例

以 DescribeRegions API 为例，假设使用的 AccessKeyId 为 testid，AccessKey Secret 为 testsecret。签名前的请求 URL 如下：

```
http://ecs.aliyuncs.com/?Timestamp=2016-02-23T12%3A46:24Z&Format=XML&AccessKeyId=testid&Action=DescribeRegions&SignatureMethod=HMAC-SHA1&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&SignatureVersion=1.0
```

使用 testsecret&，计算得到的签名值是：

```
OLeaidS1JvxuMvnyHOWuj+uX5qY=
```

最后将签名作为 Signature 参数加入到 URL 请求中，最后得到的 URL 为：

```
http://ecs.aliyuncs.com/?SignatureVersion=1.0&Action=DescribeRegions&Format=XML&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf&Version=2014-05-26&AccessKeyId=testid&Signature=OLeaidS1JvxuMvnyHOWuj+uX5qY=&SignatureMethod=HMAC-SHA1&Timestamp=2016-02-23T12%3A46%3A24Z
```

7.公共参数

本文列举了

消息队列RocketMQ版

的 API 的公共请求参数和公共返回参数。

公共请求参数

公共请求参数是每个接口都需要使用到的请求参数。

名称	类型	是否必需	描述
Format	String	否	返回消息的格式。取值：JSON XML（默认值）
productName	String	是	API 的产品名称，直接填写 Ons 即可。
domain	String	是	API 的接入点 domain，规则是 <code>ons.\${RegionId}.aliyuncs.com</code> ，该 domain 对应的地域必须和 regionId 对应的地域一致。
Version	String	是	API 版本号，使用 YYYY-MM-DD 日期格式。取值：2019-02-14
AccessKeyId	String	是	访问服务使用的密钥对。
AccessKeySecret	String	是	- 如果您以阿里云账号（主账号）或 RAM 用户（子账号）调用 API，则此参数为阿里云账号或 RAM 用户的 accessKeyId/accessKeySecret。 - 如果您以 RAM 角色调用 API，则此参数为您获取的 STS 安全令牌中的 accessKeyId/accessKeySecret。详情参见RAM主子账号授权。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1

名称	类型	是否必需	描述
Timestamp	String	是	请求的时间戳，为日期格式。使用 UTC 时间按照 ISO 8601标准，格式为 YYYY-MM-DDThh:mm:ssZ。例如，北京时间 2013 年 1 月 10 日 20 点 0 分 0 秒，表示为 2013-01-10T12:00:00Z。
SignatureVersion	String	是	签名算法版本。取值：1.0
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。在不同请求间要使用不同的随机数值。

请求示例

```
https://ons.cn-hangzhou.aliyuncs.com/?Action=OnsRegionList
&Format=JSON
&Version=2019-02-14
&AccessKeyId=key-test
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgl%3D
&SignatureMethod=HMAC-SHA1
&Timestamp=2020-01-01T12:00:00Z
&SignatureNonce=15215528852396
&SignatureVersion=1.0
...
```

公共返回参数

每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码 RequestId。调用 API 服务后返回数据采用统一格式。返回的 HTTP 状态码为 2xx，代表调用成功；返回的 HTTP 状态码为 4xx 或 5xx，代表调用失败。

XML 示例

```
<?xml version="1.0" encoding="UTF-8"?>
<!--结果的根结点-->
<接口名称+Response>
| <!--返回请求标签-->
| <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
| <!--返回结果数据-->
</接口名称+Response>
```

JSON 示例

```
{  
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216"  
  /* 返回结果数据 */  
}
```

8. 获取 AccessKey

您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云 API 时您需要使用 AccessKey 完成身份验证。

背景信息

AccessKey 包括 AccessKey ID 和 AccessKey Secret。

- AccessKeyId：用于标识用户。
- AccessKeySecret：用于验证用户的密钥。AccessKeySecret 必须保密。

 **警告** 主账号 Accesskey 泄露会威胁您所有资源的安全。建议使用子账号（RAM 用户）Accesskey 进行操作，可以有效降低 Accesskey 泄露的风险。

操作步骤

1. 以主账号登录 [阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击 **AccessKey 管理**。
3. 在安全提示页面，选择获取主账号还是子账号的 AccessKey。

4. 获取账号 Accesskey。
 - 获取主账号 AccessKey
 - a. 单击**继续使用AccessKey**。
 - b. 在**安全信息管理**页面，单击**创建AccessKey**。
 - c. 在**手机验证**页面，获取验证码，完成手机验证，单击**确定**。
 - d. 在**新建用户AccessKey**页面，展开 **AccessKey详情**，查看 AccessKeyId 和 AccessKeySecret。可以单击**保存AK信息**，下载 AccessKey 信息。

- 获取子账号 AccessKey
 - a. 单击**开始使用子用户AccessKey**。
 - b. 如果未创建 RAM 用户，在系统跳转的 **RAM访问控制台** 的**新建用户**页面，创建 RAM 用户。如果是获取已有 RAM 用户的 Accesskey，则跳过此步骤。
 - c. 在 **RAM访问控制台** 的左侧导航栏，选择**人员管理 > 用户**，搜索需要获取 AccessKey 的用户。
 - d. 单击用户登录名称，在用户详情页**认证管理**页签下的**用户AccessKey**区域，单击**创建新的 AccessKey**
 - e. 在**创建AccessKey**页面，查看 AccessKeyId 和 AccessKeySecret。可以单击**下载CSV文件**，下载 AccessKey 信息或者单击**复制**，复制 AccessKey 信息。

9. 错误码列表

如果成功调用

消息队列RocketMQ版

的 API，则返回给客户端 ResponseCode=200；如果调用失败，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	消息队列RocketMQ版 后端异常	通过阿里云工单联系技术支持人员
ONS_SERVICE_UNSUPPORTED	当前调用在对应的地域不支持	请联系技术支持人员确认该接口的开通情况
ONS_INVOKE_ERROR	OpenAPI 接口调用失败	联系技术支持人员处理
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_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

10. 服务开通接口

10.1. OpenOnsService

调用 OpenOnsService 开通消息队列 RocketMQ 版服务。

使用说明

目前，仅可通过华东1（杭州）地域开通服务。开通服务不被计费。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OpenOnsService	系统规定参数。取值： OpenOnsService 。

返回数据

名称	类型	示例值	描述
OrderId	String	2068689****0272	订单 ID。
RequestId	String	8C5B4603-8977-4513-AB60-9C3E2F88****	公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OpenOnsService
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OpenOnsServiceResponse>
  <RequestId>8C5B4603-8977-4513-AB60-9C3E2F88****</RequestId>
  <OrderId>2068689****0272</OrderId>
</OpenOnsServiceResponse>
```

JSON 格式

```
{
  "RequestId": "8C5B4603-8977-4513-AB60-9C3E2F88****",
  "OrderId": "2068689****0272"
}
```

错误码

访问[错误中心](#)查看更多错误码。

11.Region 管理接口

11.1. OnsRegionList

使用 OnsRegionList 获取消息队列 RocketMQ 版（下文简称 RocketMQ）目前开放服务的地域（Region）信息列表。

在您使用 SDK 接入 RocketMQ 服务并进行管控操作时，需要依次指定两个地域的信息，第二个地域的信息才是通过 OnsRegionList 接口获取。具体场景如下：

- **将您的应用接入 RocketMQ 服务：**根据您的应用的部署地域选择就近的 API 网关接入点，填写相应的 RegionId。该 RegionId 实际上只用于接入阿里云 OpenAPI 网关，因为 RocketMQ 通过阿里云 OpenAPI 平台（即 POP）来提供 API 服务。
- **连接要进行管控操作的地域：**指定您要对哪个地域的 RocketMQ 资源进行操作，填写该地域相应的 RegionId。此时要填写的 RegionId 即可通过 OnsRegionList 接口获取。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsRegionList	系统规定参数。取值：OnsRegionList。

返回数据

名称	类型	示例值	描述
Data	Array		查询结果。
RegionDo			
OnsRegionId	String	cn-hangzhou	地域的 ID。
RegionName	String	华东1（杭州）	地域名称。
RequestId	String	72D14A84-45E5-4E01-A6DB-F63C4721A046	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsRegionList
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsRegionListResponse>
<Data>
  <RegionDo>
    <RegionName>公网</RegionName>
    <OnsRegionId>mq-internet-access</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华北1（青岛）</RegionName>
    <OnsRegionId>cn-qingdao</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华南1（深圳）</RegionName>
    <OnsRegionId>cn-shenzhen</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华东1（杭州）</RegionName>
    <OnsRegionId>cn-hangzhou</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华北2（北京）</RegionName>
    <OnsRegionId>cn-beijing</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华东2（上海）</RegionName>
    <OnsRegionId>cn-shanghai</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>西南1（成都）</RegionName>
    <OnsRegionId>cn-chengdu</OnsRegionId>
  </RegionDo>
  <RegionDo>
    <RegionName>华北3（张家口）</RegionName>
    <OnsRegionId>cn-zhangjiakou</OnsRegionId>
  </RegionDo>
```

```
<RegionDo>
  <RegionName>华北5（呼和浩特）</RegionName>
  <OnsRegionId>cn-huhehaote</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>中国（香港）</RegionName>
  <OnsRegionId>cn-hongkong</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>新加坡</RegionName>
  <OnsRegionId>ap-south-east-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>日本（东京）</RegionName>
  <OnsRegionId>ap-north-east-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>马来西亚（吉隆坡）</RegionName>
  <OnsRegionId>ap-south-east-3</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>印度（孟买）</RegionName>
  <OnsRegionId>ap-south-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>阿联酋（迪拜）</RegionName>
  <OnsRegionId>me-east-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>德国（法兰克福）</RegionName>
  <OnsRegionId>eu-central-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>美国（硅谷）</RegionName>
  <OnsRegionId>us-west-1</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>美国（弗吉尼亚）</RegionName>
  <OnsRegionId>us-east-1</OnsRegionId>
</RegionDo>
<RegionDo>
```

```
<RegionName>澳大利亚（悉尼）</RegionName>
<OnsRegionId>ap-south-east-2</OnsRegionId>
</RegionDo>
<RegionDo>
  <RegionName>英国（伦敦）</RegionName>
  <OnsRegionId>eu-west-1</OnsRegionId>
</RegionDo>
</Data>
<RequestId>72D14A84-45E5-4E01-A6DB-F63C4721A046</RequestId>
<OnsRegionListResponse>
```

JSON 格式

```
{
  "Data": {
    "RegionDo": [
      {
        "RegionName": "公网",
        "OnsRegionId": "mq-internet-access"
      },
      {
        "RegionName": "华北1（青岛）",
        "OnsRegionId": "cn-qingdao"
      },
      {
        "RegionName": "华南1（深圳）",
        "OnsRegionId": "cn-shenzhen"
      },
      {
        "RegionName": "华东1（杭州）",
        "OnsRegionId": "cn-hangzhou"
      },
      {
        "RegionName": "华北2（北京）",
        "OnsRegionId": "cn-beijing"
      },
      {
        "RegionName": "华东2（上海）",
        "OnsRegionId": "cn-shanghai"
      },
      {
```

```
"RegionName": "西南1（成都）",
"OnsRegionId": "cn-chengdu"
},
{
  "RegionName": "华北3（张家口）",
  "OnsRegionId": "cn-zhangjiakou"
},
{
  "RegionName": "华北5（呼和浩特）",
  "OnsRegionId": "cn-huhehaote"
},
{
  "RegionName": "中国（香港）",
  "OnsRegionId": "cn-hongkong"
},
{
  "RegionName": "新加坡",
  "OnsRegionId": "ap-southeast-1"
},
{
  "RegionName": "日本（东京）",
  "OnsRegionId": "ap-northeast-1"
},
{
  "RegionName": "马来西亚（吉隆坡）",
  "OnsRegionId": "ap-southeast-3"
},
{
  "RegionName": "印度（孟买）",
  "OnsRegionId": "ap-south-1"
},
{
  "RegionName": "阿联酋（迪拜）",
  "OnsRegionId": "me-east-1"
},
{
  "RegionName": "德国（法兰克福）",
  "OnsRegionId": "eu-central-1"
},
{
  "RegionName": "美国（硅谷）"
```

```
    "RegionName": "美国（硅谷）",  
    "OnsRegionId": "us-west-1"  
  },  
  {  
    "RegionName": "美国（弗吉尼亚）",  
    "OnsRegionId": "us-east-1"  
  },  
  {  
    "RegionName": "澳大利亚（悉尼）",  
    "OnsRegionId": "ap-southeast-2"  
  },  
  {  
    "RegionName": "英国（伦敦）",  
    "OnsRegionId": "eu-west-1"  
  }  
],  
"RequestId": "72D14A84-45E5-4E01-A6DB-F63C4721A046"  
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.实例管理接口

12.1. OnsInstanceCreate

使用 OnsInstanceCreate 创建实例。

使用本接口时，请注意以下几点：

- 默认仅限主账号使用本接口，RAM 用户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 8。
- 实例名称在该实例所部署的地域下要唯一，不可重名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsInstanceCreate	系统规定参数。取值： OnsInstanceCreate 。
InstanceName	String	是	测试实例	实例名称。 长度限制为3 ~ 64 个字符，可以是中文、英文、数字、短划线（-）以及下划线（_）。
Remark	String	否	备注说明	备注。

返回数据

名称	类型	示例值	描述
Data	Struct		返回结果。
InstanceId	String	MQ_INST_188077086902****_BXSuw61e	创建的实例 ID。
InstanceType	Integer	1	创建的实例类型，取值说明如下： • 1：后付费实例 • 2：铂金版实例

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA11111****	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsInstanceCreate
&InstanceName=测试实例
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsInstanceCreateResponse>
  <Data>
    <InstanceId>MQ_INST_188077086902****_BXSUW61e</InstanceId>
    <InstanceType>1</InstanceType>
  </Data>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA11111****</RequestId>
</OnsInstanceCreateResponse>
```

JSON 格式

```
{
  "Data": {
    "InstanceId": "MQ_INST_188077086902****_BXSUW61e",
    "InstanceType": 1
  },
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA11111****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.2. OnsInstanceBaseInfo

使用 OnsInstanceBaseInfo 查询实例基本信息和收发消息的接入点。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsInstanceBaseInfo	系统规定参数。取值： OnsInstanceBaseInfo 。
InstanceId	String	是	MQ_INST_138015630679****_BAAY1Hac	需查询的实例 ID。

返回数据

名称	类型	示例值	描述
InstanceBaseInfo	Struct		实例信息对象。
Endpoints	Struct		各个协议的接入点。
HttpInternalEndpoint	String	http://138015630679****.mqrest.cn-zhangjiakou-internal.aliyuncs.com	HTTP 内网接入点。
HttpInternetEndpoint	String	http://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com	HTTP 公网接入点。
HttpInternetSecureEndpoint	String	https://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com	HTTPS 公网接入点。
TcpEndpoint	String	http://MQ_INST_138015630679****_BAAY1Hac.cn-zhangjiakou.mq-internal.aliyuncs.com:8080	TCP 协议接入点。
IndependentNaming	Boolean	true	实例是否有命名空间。取值说明如下： 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。

名称	类型	示例值	描述
InstanceId	String	MQ_INST_138015630679****_BAAY1Hac	实例 ID。
InstanceName	String	test	实例名称。 长度限制为3 ~ 64 个字符，可以是中文、英文、数字、短划线（-）以及下划线（_）。
InstanceStatus	Integer	5	实例状态。取值说明如下： 0：铂金版实例部署中 2：后付费实例已欠费 5：后付费实例或铂金版实例服务中 7：铂金版实例升级中且服务可用
InstanceType	Integer	2	实例类型。取值说明如下： 1：后付费实例 2：铂金版实例
MaxTps	Long	10000	消息收发 TPS 上限。取值如下： 5000、10000、20000、50000、100000、200000、300000、500000、800000、1000000。 详细的解释以购买页的说明为准。  说明 该参数仅适用于消息队列 RocketMQ 版企业铂金版实例。
ReleaseTime	Long	1603555200000	铂金版实例过期时间。
Remark	String	test	备注。
TopicCapacity	Integer	50	该实例下允许创建的 Topic 数量上限。取值如下： 25、50、100、300、500。  说明 该参数仅适用于消息队列 RocketMQ 版企业铂金版实例。
RequestId	String	6CC46974-65E8-4C20-AB07-D20D102E****	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-zhangjiakou.aliyuncs.com/?Action=OnsInstanceBaseInfo
&InstanceId=MQ_INST_138015630679****_BAAY1Hac
&<公共请求参数>
```

正常返回示例

XML 格式

```
<InstanceBaseInfoResponse>
  <InstanceBaseInfo>
    <InstanceName>test</InstanceName>
    <MaxTps>10000</MaxTps>
    <Endpoints>
      <HttpInternalEndpoint>http://138015630679****.mqrest.cn-zhangjiakou-internal.aliyuncs.com</HttpInternalEndpoint>
      <TcpEndpoint>http://MQ_INST_138015630679****_BAAY1Hac.cn-zhangjiakou.mq-internal.aliyuncs.com:8080</TcpEndpoint>
      <HttpInternetEndpoint>http://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com</HttpInternetEndpoint>
      <HttpInternetSecureEndpoint>https://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com</HttpInternetSecureEndpoint>
    </Endpoints>
    <TopicCapacity>50</TopicCapacity>
    <InstanceId>MQ_INST_138015630679****_BAAY1Hac</InstanceId>
    <ReleaseTime>1603555200000</ReleaseTime>
    <InstanceStatus>5</InstanceStatus>
    <IndependentNaming>true</IndependentNaming>
    <InstanceType>2</InstanceType>
    <Remark>onspre-cn-m7r1r5f****</Remark>
  </InstanceBaseInfo>
  <RequestId>6CC46974-65E8-4C20-AB07-D20D102E****</RequestId>
</InstanceBaseInfoResponse>
```

JSON 格式

```
{
  "InstanceBaseInfo": {
    "InstanceName": "test",
    "MaxTps": 10000,
    "Endpoints": {
      "HttpInternalEndpoint": "http://138015630679****.mqrest.cn-zhangjiakou-internal.aliyuncs.com",
      "TcpEndpoint": "http://MQ_INST_138015630679****_BAAY1Hac.cn-zhangjiakou.mq-internal.aliyuncs.com:8080",
      "HttpInternetEndpoint": "http://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com",
      "HttpInternetSecureEndpoint": "https://138015630679****.mqrest.cn-zhangjiakou.aliyuncs.com"
    },
    "TopicCapacity": 50,
    "InstanceId": "MQ_INST_138015630679****_BAAY1Hac",
    "ReleaseTime": 1603555200000,
    "InstanceStatus": 5,
    "IndependentNaming": true,
    "InstanceType": 2,
    "Remark": "onspre-cn-m7r1r5f****"
  },
  "RequestId": "6CC46974-65E8-4C20-AB07-D20D102E****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.3. OnsInstanceDelete

使用 OnsInstanceDelete 删除实例。

使用本接口时，请注意以下几点：

- 默认仅限主账号使用本接口，RAM 用户只有在被授予了相关权限后方可使用。
- 实例下 Topic 和 Group ID 为空时才可删除实例。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	OnsInstanceDelete	系统规定参数。取值： OnsInstanceDelete 。
InstanceId	String	是	MQ_INST_188077086902****_BXSUW61e	实例 ID。

返回数据

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA11111****	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsInstanceDelete
&InstanceId=MQ_INST_188077086902****_BXSUW61e
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsInstanceDeleteResponse>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA11111****</RequestId>
</OnsInstanceDeleteResponse>
```

JSON 格式

```
{
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA11111****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.4. OnsInstanceUpdate

使用 OnsInstanceUpdate 更新实例名称和描述。

使用本接口时，请注意以下几点：

- 默认仅限主账号使用本接口，RAM 用户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 8。
- 实例名称在该实例所部署的地域下要唯一，不可重名。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsInstanceUpdate	系统规定参数。取值： OnsInstanceUpdate 。
InstanceId	String	是	MQ_INST_188077086902****_BXSUW61e	实例 ID。
InstanceName	String	是	名字更新	实例名称。 长度限制为3 ~ 64 个字符，可以是中文、英文、数字、短划线（-）以及下划线（_）。 传空值则不更新。
Remark	String	否	备注更新	备注，传空值则不修改。

返回数据

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA11111****	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsInstanceUpdate
&InstanceId=MQ_INST_188077086902****_BXSUW61e
&InstanceName=名字更新
&Remark=备注更新
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsInstanceUpdateResponse>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111****</RequestId>
</OnsInstanceUpdateResponse>
```

JSON 格式

```
{
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA111111****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.5. OnsInstanceInServiceList

使用 OnsInstanceInServiceList 查询当前账号在某地域（Region）下所有实例的信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsInstanceInServiceList	系统规定参数。取值：OnsInstanceInServiceList。
Tag.N.Key	String	是	CartService	资源的标签键。 - N 的取值范围为 1~20。 - 一旦传入该值，则不允许为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Tag.N.Value	String	是	SericeA	资源的标签值。 - N 的取值范围为 1~20。 - 一旦传入该值，可以为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http://或者 https://。

返回数据

名称	类型	示例值	描述
Data	Array of InstanceVO		返回所有已发布的实例信息列表。
InstanceVO			
IndependentNaming	Boolean	true	实例是否有命名空间。取值说明如下： 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。
InstanceId	String	MQ_INST_188077086902****_BXSUW61e	实例 ID。
InstanceName	String	test1	实例名称。 长度限制 3~64 个字符，可以是中文、英文、数字、短划线（-）以及下划线（_）。
InstanceStatus	Integer	5	实例状态。取值说明如下： 0：铂金版实例部署中 2：后付费实例已欠费 5：后付费实例或铂金版实例服务中 7：铂金版实例升级中且服务可用
InstanceType	Integer	2	实例类型。取值说明如下： 1：后付费实例 2：铂金版实例
ReleaseTime	Long	1551024000000	铂金版实例的过期时间。
Tags	Array of Tag		实例所绑定的标签。
Tag			
Key	String	CartService	标签键。
Value	String	ServiceA	标签值。

名称	类型	示例值	描述
RequestId	String	0598E46F-DB06-40E2-AD7B-C45923EE****	公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-zhangjiakou.aliyuncs.com/?Action=OnsInstanceInServiceList
&Tag.1.Key=CartService
&Tag.1.Value=ServiceA
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsInstanceInServiceListResponse>
  <RequestId>0598E46F-DB06-40E2-AD7B-C45923EE****</RequestId>
  <Data>
    <InstanceVO>
      <InstanceName>test1</InstanceName>
      <InstanceId>MQ_INST_188077086902****_BXSUW61e</InstanceId>
      <ReleaseTime>1551024000000</ReleaseTime>
      <InstanceStatus>5</InstanceStatus>
      <IndependentNaming>true</IndependentNaming>
      <InstanceType>2</InstanceType>
      <Tags>
        <Tag>
          <Value>ServiceA</Value>
          <Key>CartService</Key>
        </Tag>
      </Tags>
    </InstanceVO>
  </Data>
</OnsInstanceInServiceListResponse>
```

JSON 格式

```
{
  "RequestId": "0598E46F-DB06-40E2-AD7B-C45923EE****",
  "Data": {
    "InstanceVO": [ {
      "InstanceName": "test1",
      "Instanceld": "MQ_INST_188077086902****_BXSuw61e",
      "ReleaseTime": 1551024000000,
      "InstanceStatus": 5,
      "IndependentNaming": true,
      "InstanceType": 2,
      "Tags": {
        "Tag": [
          {
            "Value": "ServiceA",
            "Key": "CartService"
          }
        ]
      }
    }
  ]
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.Topic 管理接口

13.1. OnsTopicCreate

使用 OnsTopicCreate 创建 Topic。

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，RAM 子账号只有在被授予了相关权限后方可使用。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicCreate	系统规定参数。取值：OnsTopicCreate。
InstanceId	String	是	MQ_INST_111111 111111_DOxxxxxx	需创建的 Topic 所对应的实例 ID。
MessageType	Integer	是	0	消息类型。取值说明如下： 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
Topic	String	是	test	需创建的 Topic 的名称： “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。
Remark	String	否	测试	备注。

返回数据

名称	类型	示例值	描述
RequestId	String	B6949B58-223E-4B75-B4FE-7797C15E2D1F	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicCreate
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&MessageType=0
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicCreateResponse>
  <RequestId>B6949B58-223E-4B75-B4FE-7797C15E2D1F</RequestId>
</OnsTopicCreateResponse>
```

JSON 格式

```
{
  "RequestId": "B6949B58-223E-4B75-B4FE-7797C15E2D1F"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.2. OnsTopicDelete

使用 OnsTopicDelete 删除您账号下指定的 Topic。

OnsTopicDelete 接口一般是在应用下线等资源回收场景下使用。Topic 删除时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。

使用 **OnsTopicDelete** 接口时，请注意以下几点：

- 默认仅限主账号使用本接口，RAM 子账号只有在被授予了相关权限后方可使用。
- 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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicDelete	系统规定参数。取值：OnsTopicDelete。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需删除的 Topic 所对应的实例 ID。
Topic	String	是	test	指定删除的 Topic。

返回数据

名称	类型	示例值	描述
RequestId	String	4189D4A6-231A-4028-8D89-F66A76C1F4F0	为公共参数，每个请求的 ID 都是唯一的，可用于排查和定位问题。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicDelete
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicDeleteResponse>
  <RequestId>4189D4A6-231A-4028-8D89-F66A76C1F4F0</RequestId>
</OnsTopicDeleteResponse>
```

JSON 格式

```
{
  "RequestId": "4189D4A6-231A-4028-8D89-F66A76C1F4F0"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.3. OnsTopicStatus

使用 OnsTopicStatus 查询当前 Topic 下的消息总量以及 Topic 的最后更新时间。

本接口一般用于判断 Topic 资源的使用率。调用本接口后返回的信息包含当前服务器上该 Topic 所有的消息数，以及最后消息写入时间。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicStatus	系统规定参数。取值：OnsTopicStatus。
Instanceld	String	是	MQ_INST_111111 111111_DOxxxxxx	需查询的 Topic 所对应的实例 ID。
Topic	String	是	test	需查询的 Topic 名称。

返回数据

名称	类型	示例值	描述
Data	Struct		查询到的 Topic 的信息存储数据结构。
LastTimeStamp	Long	1570864984364	当前 Topic 的最后更新时间。
Perm	Integer	6	所有关系编号。取值说明如下： 2：可以发布 4：可以订阅 6：可以发布和订阅
TotalCount	Long	2310	当前 Topic 的所有分区存在的消息数总和。
RequestId	String	427EE49D-D762- 41FB-8F3D- 9BAC96C314B4	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicStatus
&InstanceId=MQ_INST_11111111111_DOxxxxxx
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicStatusResponse>
  <Data>
    <Perm>6</Perm>
    <TotalCount>2310</TotalCount>
    <LastTimeStamp>1570864984364</LastTimeStamp>
  </Data>
  <RequestId>427EE49D-D762-41FB-8F3D-9BAC96C314B4</RequestId>
</OnsTopicStatusResponse>
```

JSON 格式

```
{
  "Data": {
    "Perm": 6,
    "TotalCount": 2310,
    "LastTimeStamp": 1570864984364
  },
  "RequestId": "427EE49D-D762-41FB-8F3D-9BAC96C314B4"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.4. OnsTopicList

使用 OnsTopicList 查询账号下所有 Topic 的信息列表。

本接口一般用于生成资源列表的场景，并不查看具体信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicList	系统规定参数。取值：OnsTopicList。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需获取的 Topic 信息列表所对应的实例 ID。
Tag.N.Key	String	是	CartService	资源的标签键。 - N 的取值范围为 1~20。 - 一旦传入该值，则不允许为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Tag.N.Value	String	是	ServiceA	资源的标签值。 - N 的取值范围为 1~20。 - 一旦传入该值，可以为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Topic	String	否	test	查询指定 Topic 时设置，否则查询所有 Topic。

返回数据

名称	类型	示例值	描述
Data	Array		返回所有已发布的 Topic 信息列表。
PublishInfoDo			
CreateTime	Long	1570700947000	创建时间。
IndependentNaming	Boolean	true	Topic 所在实例是否有命名空间。取值说明如下： 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。
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	Topic 所在的实例 ID。

名称	类型	示例值	描述
MessageType	Integer	0	消息类型。取值说明如下： 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
Owner	String	138015630679****	Topic 所有者编号，为阿里云的账号 ID。
Relation	Integer	6	所有关系编号。取值说明如下： 1：持有者 2：可以发布 4：可以订阅 6：可以发布和订阅
RelationName	String	授权发布/订阅	所有关系名称，例如持有者、可订阅、可发布、可发布订阅。
Remark	String	测试	备注信息。
Tags	Array		Topic 所绑定的标签。
Tag			
Key	String	CartService	标签键。
Value	String	SrviceA	标签值。
Topic	String	test	Topic 名称。
RequestId	String	4A978869-7681-4529-B470-107E1379F8F2	为公共参数，每个请求 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicList
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Tag.1.Key=CartService
&Tag.1.Value=ServiceA
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicListResponse>
  <RequestId>4A978869-7681-4529-B470-107E1379FBF2</RequestId>
  <Data>
    <PublishInfoDo>
      <Owner>138015630679****</Owner>
      <Relation>6</Relation>
      <InstanceId>MQ_INST_111111111111_DOxxxxxx</InstanceId>
      <CreateTime>1570700947000</CreateTime>
      <RelationName>授权发布/订阅</RelationName>
      <IndependentNaming>true</IndependentNaming>
      <Topic>test</Topic>
      <MessageType>0</MessageType>
      <Remark>测试</Remark>
      <Tags>
        <Tag>
          <Value>SrvceA</Value>
          <Key>Cart Service</Key>
        </Tag>
      </Tags>
    </PublishInfoDo>
  </Data>
</OnsTopicListResponse>
```

JSON 格式

```
{
  "RequestId": "4A978869-7681-4529-B470-107E1379FBF2",
  "Data": {
    "PublishInfoDo": [
      {
        "Owner": "138015630679****",
        "Relation": 6,
        "Instanceld": "MQ_INST_111111111111_DOxxxxxx",
        "CreateTime": 1570700947000,
        "RelationName": "授权发布/订阅",
        "IndependentNaming": true,
        "Topic": "test",
        "MessageType": 0,
        "Remark": "测试",
        "Tags": {
          "Tag": [
            {
              "Value": "SrvceA",
              "Key": "CartService"
            }
          ]
        }
      }
    ]
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.5. OnsTopicUpdate

使用 OnsTopicUpdate 配置 Topic 的读写模式。

默认仅限主账号使用本接口，RAM 子账号只有在被授予了相关权限后方可使用。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicUpdate	系统规定参数。取值： OnsTopicUpdate。
InstanceId	String	是	MQ_INST_111111 111111_DOxxxxxx	需配置的 Topic 所对应的实例 ID。
Perm	Integer	是	6	设置该 Topic 的读写模式。取值说明如下： • 6：同时支持读写 • 4：禁写 • 2：禁读
Topic	String	是	test	需配置的 Topic 名称，在一个用户下不可重复，即使地域（Region）不同。

返回数据

名称	类型	示例值	描述
RequestId	String	81979ADA-4A78- 4F64-9DEC- 5700446DF952	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicUpdate
&Perm=6
&Topic=test
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicUpdateResponse>
  <RequestId>81979ADA-4A78-4F64-9DEC-5700446DF952</RequestId>
</OnsTopicUpdateResponse>
```

JSON 格式

```
{
  "RequestId": "81979ADA-4A78-4F64-9DEC-5700446DF952"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.6. OnsTopicSubDetail

一般使用 OnsTopicSubDetail 查看有哪些在线订阅组订阅了这个 Topic。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTopicSubDetail	系统规定参数。取值：OnsTopicSubDetail。
InstanceId	String	是	MQ_INST_1111111_111111_DOxxxxxx	需查询的 Topic 所对应的实例 ID。
Topic	String	是	test	需查询的 Topic。

返回数据

名称	类型	示例值	描述
Data	Struct		返回 Data 对象。
SubscriptionDataList	Array		所有在线订阅组。
SubscriptionDataList			
GroupId	String	GIID_test	订阅该 Topic 的 Group ID。
MessageModel	String	CLUSTERING	消费模式。取值说明如下： - CLUSTERING：集群订阅 - BROADCASTING：广播订阅

名称	类型	示例值	描述
SubString	String	*	订阅表达式。
Topic	String	test	Topic 名称。
RequestId	String	87B6207F-2908-42B5-A134-84956DCABF3D	为公共参数，每个请求 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTopicSubDetail
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTopicSubDetailResponse>
  <Data>
    <Topic>test</Topic>
    <SubscriptionDataList>
      <SubscriptionDataList>
        <SubString>*</SubString>
        <MessageModel>CLUSTERING</MessageModel>
        <GroupId>GID_test</GroupId>
      </SubscriptionDataList>
    </SubscriptionDataList>
  </Data>
  <RequestId>87B6207F-2908-42B5-A134-84956DCABF3D</RequestId>
</OnsTopicSubDetailResponse>
```

JSON 格式

```
{
  "Data": {
    "Topic": "test",
    "SubscriptionDataList": {
      "SubscriptionDataList": [
        {
          "SubString": "**",
          "MessageModel": "CLUSTERING",
          "GroupId": "GID_test"
        }
      ]
    }
  },
  "RequestId": "87B6207F-2908-42B5-A134-84956DCABF3D"
}
```

错误码

访问[错误中心](#)查看更多错误码。

14.Group 管理接口

14.1. OnsGroupCreate

使用 OnsGroupCreate 创建客户端 Group ID。
创建的该 Group ID 用于发布和订阅消息。详情请参见[名词解释](#)。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsGroupCreate	系统规定参数。取值：OnsGroupCreate。
GroupId	String	是	GID_test_groupId	创建的消息消费集群的 Group ID。取值说明如下： - 以 “GID_ ” 或者 “GID- ” 开头，只能包含字母、数字、短横线（-）和下划线（_）。 - 长度限制在 5-64 字节之间。  说明 Group ID 一旦创建，将无法再修改。
InstanceId	String	是	MQ_INST_111111 111111_DOxxxxxx	需创建的 Group ID 所对应的实例 ID。
Remark	String	否	test	备注。
GroupType	String	否	tcp	指定创建的 Group ID 适用的协议。TCP 协议和 HTTP 协议的 Group ID 不可以共用，需要分别创建。取值说明如下： tcp：默认值，表示创建的 Group ID 仅适用于 TCP 协议的消息收发。 http：表示创建的 Group ID 仅适用于 HTTP 协议的消息收发。

返回数据

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsGroupCreate
&GroupId=GID_test_groupId
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Remark=test
&GroupType=tcp
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsGroupCreateResponse>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsGroupCreateResponse>
```

JSON 格式

```
{
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

14.2. OnsGroupDelete

使用 OnsGroupDelete 删除之前使用 OnsGroupCreate 接口创建的 Group。

OnsGroupCreate 接口的详细信息，请参见 [OnsGroupCreate](#)。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsGroupDelete	系统规定参数。取值：OnsGroupDelete。
GroupId	String	是	GD_test_groupId	需要删除的消息消费集群的 Group ID。
InstanceId	String	是	MQ_INST_111111 111111_DOxxxxxx	需删除的 Group ID 对应的实例 ID。

返回数据

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsGroupDelete
&GroupId=GD_test_groupId
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsGroupDeleteResponse>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsGroupDeleteResponse>
```

JSON 格式

```
{
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

14.3. OnsGroupList

使用 OnsGroupList 展示您的 Group ID 资源的列表，并不用于具体的信息查询。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsGroupList	系统规定参数。取值：OnsGroupList。
InstanceId	String	是	MQ_INST_111111 111111_DOxxxxxx	需获取资源列表的 Group ID 所对应的实例 ID。
Tag.N.Key	String	是	CartService	资源的标签键。 - N 的取值范围为 1~20。 - 一旦传入该值，则不允许为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Tag.N.Value	String	是	ServiceA	资源的标签值。 - N 的取值范围为 1~20。 - 一旦传入该值，可以为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
GroupId	String	否	GlD_test_group_i d	查询指定 Group ID 时设置，否则查询所有 Group ID。
GroupType	String	否	tcp	查询的 Group ID 适用的协议。TCP 协议和 HTTP 协议的 Group ID 不可以共用，需要分别创建。取值说明如下： tcp：默认值，表示该 Group ID 仅适用于 TCP 协议的消息收发。 http：表示该 Group ID 仅适用于 HTTP 协议的消息收发。

返回数据

名称	类型	示例值	描述
Data	Array		查询到的订阅关系集合。
SubscribeInfoDo			
CreateTime	Long	1568896605000	Group ID 的创建时间。
GroupId	String	GID_test_group_id	消费集群组 ID。
GroupType	String	tcp	查询的 Group ID 适用的协议。TCP 协议和 HTTP 协议的 Group ID 不可以共用，需要分别创建。取值说明如下： tcp：表示该 Group ID 仅适用于 TCP 协议的消息收发。 http：表示该 Group ID 仅适用于 HTTP 协议的消息收发。
IndependentNaming	Boolean	true	实例是否有命名空间。取值说明如下： 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
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
Owner	String	138015630679****	该发布信息的拥有者。
Remark	String	test	备注。
Tags	Array		Group ID 所绑定的标签。
Tag			
Key	String	CartService	标签键。
Value	String	ServiceA	标签值。
UpdateTime	Long	1570700979000	该 Group ID 的更新时间。

名称	类型	示例值	描述
RequestId	String	16996623-AC4A-43AF-9248-FD9D2D75C8E8	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsGroupList
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Tag.1.Key=CartService
&Tag.1.Value=ServiceA
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsGroupListResponse>
  <RequestId>16996623-AC4A-43AF-9248-FD9D2D75C8E8</RequestId>
  <Data>
    <SubscribeInfoDo>
      <Owner>138015630679****</Owner>
      <GroupType>tcp</GroupType>
      <InstanceId>MQ_INST_111111111111_DOxxxxxx</InstanceId>
      <CreateTime>1568896605000</CreateTime>
      <UpdateTime>1570700979000</UpdateTime>
      <IndependentNaming>true</IndependentNaming>
      <GroupId>GID_test_group_id</GroupId>
      <Remark>test</Remark>
      <Tags>
        <Tag>
          <Value>ServiceA</Value>
          <Key>CartService</Key>
        </Tag>
      </Tags>
    </SubscribeInfoDo>
  </Data>
</OnsGroupListResponse>
```

JSON 格式

```
{
  "RequestId": "16996623-AC4A-43AF-9248-FD9D2D75C8E8",
  "Data": {
    "SubscribeInfoDo": [
      {
        "Owner": "138015630679****",
        "GroupType": "tcp",
        "Instanceld": "MQ_INST_111111111111_DOxxxxxx",
        "CreateTime": 1568896605000,
        "UpdateTime": 1570700979000,
        "IndependentNaming": true,
        "GroupId": "GID_test_group_id",
        "Remark": "test",
        "Tags": {
          "Tag": [
            {
              "Value": "ServiceA",
              "Key": "CartService"
            }
          ]
        }
      }
    ]
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

14.4. OnsGroupSubDetail

使用 OnsGroupSubDetail 查看 Group 订阅了哪些 Topic，如果 Group ID 对应的消费者实例不在线则查不到数据。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsGroupSubDetail	系统规定参数。取值：OnsGroupSubDetail。
GroupId	String	是	GID_test_group_id	需查询的 Group ID。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需查询的 Group ID 所属的实例 ID。

返回数据

名称	类型	示例值	描述
Data	Struct		返回 Data 对象。
GroupId	String	GID_test_group_id	需查询的 Group ID。
MessageModel	String	CLUSTERING	消费模式。取值说明如下： - CLUSTERING：集群订阅 - BROADCASTING：广播订阅
Online	Boolean	true	消费者是否在线。
SubscriptionDataList	Array		如果 Group ID 所对应的消费者不在线则查不到数据。
SubscriptionDataList			
SubString	String	*	订阅表达式。
Topic	String	test-mq_topic	订阅的 Topic。
RequestId	String	3364E875-013B-442A-BC3C-C1A84DC60834	为公共参数，每个请求 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsGroupSubDetail
&GroupId=GID_test_group_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsGroupSubDetailResponse>
  <Data>
    <MessageModel>CLUSTERING</MessageModel>
    <Online>true</Online>
    <SubscriptionDataList>
      <SubscriptionDataList>
        <SubString>*</SubString>
        <Topic>test-mq_topic</Topic>
      </SubscriptionDataList>
    </SubscriptionDataList>
    <GroupId>GID_test_group_id</GroupId>
  </Data>
  <RequestId>3364E875-013B-442A-BC3C-C1A84DC60834</RequestId>
</OnsGroupSubDetailResponse>
```

JSON 格式


```
{
  "Data": {
    "MessageModel": "CLUSTERING",
    "Online": true,
    "SubscriptionDataList": {
      "SubscriptionDataList": [
        {
          "SubString": "*",
          "Topic": "test-mq_topic"
        }
      ]
    },
    "GroupId": "GID_test_group_id"
  },
  "RequestId": "3364E875-013B-442A-BC3C-C1A84DC60834"
}
```

错误码

访问[错误中心](#)查看更多错误码。

14.5. OnsGroupConsumerUpdate

使用 **OnsGroupConsumerUpdate** 配置指定 Group ID 对应的消费集群的消息读取权限。

您可以使用本接口根据您指定的消息队列 Apache RocketMQ 版（简称 RocketMQ）地域（Region）以及 Group ID 名称，配置该 Group ID 读取消息的开关。本接口一般用于禁止特定 Group ID 读取消息的场景。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsGroupConsumerUpdate	系统规定参数。取值：OnsGroupConsumerUpdate。
GroupId	String	是	GID_test_groupId	需配置的 Group ID。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需配置的 Group ID 所对应的实例 ID。

名称	类型	是否必选	示例值	描述
ReadEnable	Boolean	是	true	设置读消息开关。

返回数据

名称	类型	示例值	描述
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsGroupConsumerUpdate
&GroupId=GID_test_groupId
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&ReadEnable=true
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsGroupConsumerUpdateResponse>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsGroupConsumerUpdateResponse>
```

JSON 格式

```
{
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

15. 标签管理接口

15.1. TagResources

使用 TagResources 为资源绑定标签。

 说明 标签的控制台操作详情请参见[标签概述](#)。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	TagResources	系统规定参数。取值： TagResources 。
ResourceId.N	RepeatList	是	TopicA	需要打标的资源 ID。资源 ID 可以是： - 实例 ID - Topic - Group ID
ResourceType	String	是	TOPIC	资源类型。枚举类型。取值： INSTANCE TOPIC GROUP
Tag.N.Key	String	是	FinanceDept	资源的标签键。 - N 的取值范围为 1~20。 - 一旦传入该值，则不允许为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Tag.N.Value	String	是	FinanceJoshua	资源的标签值。 - N 的取值范围为 1~20。 - 该值可以为空字符串。 - 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。

名称	类型	是否必选	示例值	描述
InstanceId	String	否	MQ_INST_188077086902****_BXSUW61e	所需绑定标签的资源所在的实例的 ID。  说明 如需为 Topic 或 Group ID 绑定标签，则必须填写该参数。

返回数据

名称	类型	示例值	描述
RequestId	String	301D2CBE-66F8-403D-AEC0-82582478****	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-zhangjiakou.aliyuncs.com/?Action=TagResources
&ResourceId.1=TopicA
&ResourceType=TOPIC
&Tag.1.Key=FinanceDept
&Tag.1.Value=FinanceJoshua
&InstanceId=MQ_INST_188077086902****_BXSUW61e
&<公共请求参数>
```

正常返回示例

XML 格式

```
<TagResourcesResponse>
  <RequestId>301D2CBE-66F8-403D-AEC0-82582478****</RequestId>
</TagResourcesResponse>
```

JSON 格式

```
{
  "RequestId": "301D2CBE-66F8-403D-AEC0-82582478****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

15.2. ListTagResources

调用 ListTagResources 查询资源绑定的标签列表。

 **说明** 选择 Tag.N.Key、Tag.N.Value 和 ResourceId.N 中的一个作为已知参数，可以查询另外两个。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ListTagResources	系统规定参数。取值： ListTagResources 。
ResourceType	String	是	TOPIC	资源类型。枚举类型。取值如下： • INSTANCE • TOPIC • GROUP
InstanceId	String	否	MQ_INST_188077086902****_BXSuW61e	需查询的标签所绑定的资源所在的实例的ID。  说明 如果您需查询的是 Topic、Group ID 的标签，则必须填写该参数。
Tag.N.Key	String	否	CartService	资源的标签键。 • N 的取值范围为 1~20。 • 一旦传入该值，则不允许为空字符串。 • 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
Tag.N.Value	String	否	ServiceA	资源的标签值。 • N 的取值范围为 1~20。 • 一旦传入该值，可以为空字符串。 • 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。

名称	类型	是否必选	示例值	描述
ResourceId.N	RepeatList	否	TopicA	需要打标的资源 ID。支持打标的资源包括实例、Topic、Group ID。
NextToken	String	否	caeba0****be03f84eb48b699f0a4883	下一页查询开始的 Token。

返回数据

名称	类型	示例值	描述
NextToken	String	caeba0****be03f84eb48b699f0a4883	下一页查询开始的 Token。
RequestId	String	301D2CBE-66F8-403D-AEC0-82582478****	为公共参数，每个请求的 ID 都是唯一的。
TagResources	Array of TagResource		由资源及其标签组成的集合，包含了资源 ID、资源类型和标签键值等信息。
InstanceId	String	MQ_INST_188077086902****_BXSUW61e	实例 ID。
ResourceId	String	TopicA	资源 ID。
ResourceType	String	ALIYUN::MQ::TOPIC	资源类型。
TagKey	String	CartService	标签键。
TagValue	String	ServiceA	标签值。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=ListTagResources
&ResourceType=TOPIC
&ResourceId.1=TopicA
&InstanceId=MQ_INST_188077086902****_BXSUW61e
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListTagResourcesResponse>
  <RequestId>301D2CBE-66F8-403D-AEC0-82582478****</RequestId>
  <NextToken>caeba0****be03f84eb48b699f0a4883</NextToken>
  <TagResources>
    <ResourceId>TopicA</ResourceId>
    <InstanceId>MQ_INST_188077086902****_BXSUW61e</InstanceId>
    <TagKey>CartService</TagKey>
    <ResourceType>ALIYUN::MQ::TOPIC</ResourceType>
    <TagValue>ServiceA</TagValue>
  </TagResources>
</ListTagResourcesResponse>
```

JSON 格式

```
{
  "RequestId": "301D2CBE-66F8-403D-AEC0-82582478****",
  "NextToken": "caeba0****be03f84eb48b699f0a4883",
  "TagResources": {
    "ResourceId": "TopicA",
    "InstanceId": "MQ_INST_188077086902****_BXSUW61e",
    "TagKey": "CartService",
    "ResourceType": "ALIYUN::MQ::TOPIC",
    "TagValue": "ServiceA"
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

15.3. UntagResources

使用 UntagResources 为资源解绑并删除标签。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UntagResources	系统规定参数。取值：UntagResources。
ResourceId.N	RepeatList	是	TopicA	需要解绑标签的资源 ID。
ResourceType	String	是	TOPIC	资源类型。枚举类型，目前支持的资源类型： • INSTANCE • TOPIC • GROUP
InstanceId	String	否	MQ_INST_111111_111111_DOxxxxxx	如需解除 Topic 和 Group ID 与标签的绑定，该参数为必填。
TagKey.N	RepeatList	否	CartService	资源的标签键。 • N 的取值范围为 1~20。 • 一旦传入该值，则不允许为空字符串。 • 最多支持 128 个字符，不能以 aliyun 和 acs: 开头，不能包含 http:// 或者 https://。
All	Boolean	否	false	是否删除资源绑定的全部标签。当请求中未设置 TagKey.N 时该参数有效。默认值 false。

返回数据

名称	类型	示例值	描述
RequestId	String	19780F2E-7841-4E0F-A5D9-C64A05301A2F	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=UntagResources
&ResourceId.1=TopicA
&ResourceType=TOPIC
&<公共请求参数>
```

正常返回示例

XML 格式

```
<UntagResourcesResponse>
  <RequestId>19780F2E-7841-4E0F-A5D9-C64A05301A2F</RequestId>
</UntagResourcesResponse>
```

JSON 格式

```
{"RequestId": "19780F2E-7841-4E0F-A5D9-C64A05301A2F"}
```

错误码

访问[错误中心](#)查看更多错误码。

16.消费管理接口

16.1. OnsConsumerAccumulate

使用 OnsConsumerAccumulate 查询指定 Group ID 的消息消费堆积情况，包括当前消息堆积数量和消费延迟时间等。

消费堆积查询一般在生产环境中需要关注 Group ID 消费进度时使用，用于粗略判断消息消费情况和延迟情况。您不仅可以查到该 Group ID 订阅的所有 Topic 的消息堆积情况，还可以查到每个 Topic 各自的消息堆积情况。

若您要查看该 Group ID 下各个在线消费者实例的消费信息，请使用 [OnsConsumerStatus](#) 接口查询消费状态。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsConsumerAccumulate	系统规定参数。取值：OnsConsumerAccumulate。
GroupId	String	是	GID_test_consumer_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_1111111_111111_DOxxxxxx	实例 ID。
Detail	Boolean	否	true	是否查询各个 Topic 的详细信息。取值说明如下： true：要查询详细信息，返回 DetailInTopicList 参数的详细信息。 false：不查询详细信息（默认值），DetailInTopicList 参数返回值为空。

返回数据

名称	类型	示例值	描述
Data	Struct		指定 Consumer 的消费堆积情况。
ConsumeTps	Float	10	该 Group ID 下消费者实例群组接收消息的总 TPS。

名称	类型	示例值	描述
DelayTime	Long	10000	延迟时间。
DetailInTopicList	Array		各个 Topic 具体情况。如果请求参数 Detail 取值为 false ，则该返回参数的返回值为空。
DetailInTopicDo			
DelayTime	Long	10000	该 Topic 中消息消费的最大延迟时间。
LastTimestamp	Long	1566231000000	该 Topic 中最近消费的某条消息的生产时间。
Topic	String	test-mq-topic	Topic 名称。
TotalDiff	Long	100	该 Topic 的消费堆积数量。
LastTimestamp	Long	1566231000000	该 Group ID 下消费者实例群组中最近消费的某条消息的生产时间。
Online	Boolean	true	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： true：状态为在线 false：状态为不在线
TotalDiff	Long	100	所有 Topic 的消费堆积数量。
RequestId	String	CE817BFF-B389-43CD-9419-95011AC9D5E8	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsConsumerAccumulate
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&GroupId=GID_test_consumer_id
&Detail=true
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsConsumerAccumulateResponse>
  <Data>
    <DetailInTopicList>
      <DetailInTopicDo>
        <TotalDiff>100</TotalDiff>
        <LastTimestamp>1566231000000</LastTimestamp>
        <DelayTime>10000</DelayTime>
        <Topic>test-mq-topic</Topic>
      </DetailInTopicDo>
    </DetailInTopicList>
    <TotalDiff>100</TotalDiff>
    <LastTimestamp>1566231000000</LastTimestamp>
    <DelayTime>10000</DelayTime>
    <Online>true</Online>
    <ConsumeTps>10</ConsumeTps>
  </Data>
  <RequestId>0CCF6437-CBB0-4378-BFEC-E08AC2583589</RequestId>
</OnsConsumerAccumulateResponse>
```

JSON 格式

```
{
  "Data": {
    "DetailInTopicList": {
      "DetailInTopicDo": [
        {
          "TotalDiff": 100,
          "LastTimestamp": 1566231000000,
          "DelayTime": 10000,
          "Topic": "test-mq-topic"
        }
      ]
    },
    "TotalDiff": 100,
    "LastTimestamp": 1566231000000,
    "DelayTime": 10000,
    "Online": true,
    "ConsumeTps": 10
  },
  "RequestId": "0CCF6437-CBB0-4378-BFEC-E08AC2583589"
}
```

错误码

访问[错误中心](#)查看更多错误码。

16.2. OnsConsumerStatus

使用 `OnsConsumerStatus` 接口查询指定 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 `OnsConsumerStatus` 接口时，请注意以下几点：

- `OnsConsumerStatus` 接口一般用于在粗略判断消费堆积以及客户端在线状态后，需要排查消费异常原因的场景，可以根据返回结果判断出指定 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。
- `OnsConsumerStatus` 接口需要调用大量后端接口以完成数据聚合，因此查询速度慢，不建议频繁调用。

调试

您可以在 [OpenAPI Explorer](#) 中直接运行该接口，免去您计算签名的困扰。运行成功后，[OpenAPI Explorer](#) 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsConsumerStatus	系统规定参数。取值：OnsConsumerStatus。
GroupId	String	是	GID_test_group_id	需要查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需查询的 Group ID 所对应的实例 ID。
Detail	Boolean	否	true	是否查询详细信息。取值说明如下： • true：要查询详细信息，即可获取返回参数 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	true	是否打印 Jstack 信息。取值说明如下： • true：要打印 Jstack 信息，可在 Jstack 返回参数中获取堆栈信息。  说明 如需打印 Jstack 信息，请确保您的入参 Detail 设置为 true。 • false：默认值，不打印 Jstack 信息，Jstack 返回参数为空。

返回数据

名称	类型	示例值	描述
Data	Struct		查询结果。
ConnectionSet	Array		该集群当前在线客户端信息。
ConnectionDo			
ClientAddr	String	30.5.121.**	该消费实例的地址和端口。

名称	类型	示例值	描述
ClientId	String	30.5.121.**@25560#-1999745829#-1737591554#458773089270275	消费实例的 ID。
Language	String	JAVA	消费端语言。
RemoteIP	String	42.120.74.**	宿主机 IP 地址或公网 IP 地址。
Version	String	V4_3_6_SNAPSHOT	消费端版本。
ConsumeModel	String	CLUSTERING	消费模型。取值说明如下： • CLUSTERING：集群消费模式 • BROADCASTING：广播消费模式 两种订阅模式的详情请参见 集群消费和广播消费 。
ConsumeTps	Float	0	总消费 TPS。
ConsumerConnectionInfoList	Array		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 请求参数设置为 true 。否则返回值为空。
ConsumerConnectionInfoDo			
ClientId	String	30.5.**.**@25560#-1999745829#-1737591554#458773089270275	消费实例的 ID。
Connection	String	**	连接信息。

名称	类型	示例值	描述
ConsumeModel	String	CLUSTERING	消费模型。取值说明如下： • CLUSTERING：集群消费模式 • BROADCASTING：广播消费模式 两种订阅模式的详情请参见 集群消费和广播消费 。
ConsumeType	String	PUSH	消费者消费消息的模式。取值说明如下： • PUSH：消息队列 RocketMQ 版服务端主动向消费者推送消息。 • PULL：消费者主动向消息队列 RocketMQ 版服务端拉取消息。
Jstack	Array		Jstack 堆栈信息。如需获取 Jstack 信息，请确保将 NeedJstack 入参设置为 true 。否则返回值为空。
ThreadTrackDo			
Thread	String	ConsumeMessageThread_0	线程名称。
TrackList	List	TID: 52 STATE: WAITING	Jstack 堆栈信息字符串。
Language	String	JAVA	客户端语言。
LastTimeStamp	Long	1570701368114	最后消费时间。
RunningDataList	Array		实时状态统计。
ConsumerRunningDataDo			
FailedCountPerHour	Long	0	每小时内消费失败的消息数统计。
FailedTps	Float	0	消费消息失败的 TPS 统计。
GroupId	String	0	订阅方的 Group ID。
OkTps	Float	0	消费消息成功的 TPS 统计。

名称	类型	示例值	描述
Rt	Float	0	消费 RT 时间，单位 ms。
Topic	String	test-mq_topic	订阅的 Topic 名称。
StartTimeStamp	Long	1570701361528	消费开始时间。
SubscriptionSet	Array		订阅关系集合。
SubscriptionData			
SubString	String	*	订阅该 Topic 的子类别 Tag 表达式。
SubVersion	Long	1570701364301	订阅关系版本号，为自增 Long 型。
TagsSet	List	ff	订阅的 Tag 集合。
Topic	String	test-mq_topic	订阅的 Topic 名称。
ThreadCount	Integer	20	消费线程数。
Version	String	V4_3_6	客户端版本号。
DelayTime	Long	100857	延迟时间。
DetailInTopicList	Array		各个 Topic 的消费情况。如需获取详细信息，请确保将 Detail 请求参数设置为 true 。否则返回值为空。
DetailInTopicDo			
DelayTime	Long	0	延迟时间。
LastTimestamp	Long	1570701259403	最后消费时间。
Topic	String	test-mq_topic	Topic 名称。

名称	类型	示例值	描述
TotalDiff	Long	0	该 Topic 消费总堆积数。
Instanceld	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
LastTimestamp	Long	1566883844954	最后消费时间。
Online	Boolean	true	是否在线。
RebalanceOK	Boolean	true	客户端 Rebalance 是否正常。取值说明如下： • true：正常 • false：不正常
SubscriptionSame	Boolean	true	订阅关系是否一致。
TotalDiff	Long	197	集群总的消费堆积。
RequestId	String	10EDC518-10E7-4B34-92FB-171235FA6E38	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsConsumerStatus
&GroupId=GID_test_group_id
&Instanceld=MQ_INST_111111111111_DOxxxxxx
&NeedJstack=true
&Detail=true
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsConsumerStatusResponse>
  <data>
    <connectionSet>
      <bizVersion>V4_3_6</bizVersion>
      <clientAddr>30.5.***.*</clientAddr>
      <clientId>30.5.***.*@97730#-1999745829#-1737591554#729272961762836</clientId>
```

```

    <language>JAVA</language>
    <version>V4_3_6</version>
  </connectionSet>
  <consumeModel>CLUSTERING</consumeModel>
  <consumeTps>0</consumeTps>
  <consumerConnectionInfoList>
    <bizVersion>V4_3_6</bizVersion>
    <clientId>30.5.***.*@97730#-1999745829#-1737591554#729272961762836</clientId>
    <consumeType>PUSH</consumeType>
    <jstack>
      <thread>ConsumeMessageThread_4</thread>
      <trackList>TID: 44 STATE: WAITING</trackList>
      <trackList>sun.misc.Unsafe.park(Native Method)</trackList>
      <trackList>java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)</trackList>
      <trackList>java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(Abstract
QueuedSynchronizer.java:2039)</trackList>
      <trackList>java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)</trackList>
    </jstack>
    <trackList>java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)</trackList>
    <trackList>java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)</trackList>
    <trackList>java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)</trackList>
    <trackList>java.lang.Thread.run(Thread.java:748)</trackList>
  </jstack>
  <language>JAVA</language>
  <lastTimeStamp>1570701368114</lastTimeStamp>
  <runningDataList>
    <failedCountPerHour>0</failedCountPerHour>
    <failedTps>0</failedTps>
    <okTps>0</okTps>
    <rt>0</rt>
    <topic>test-mq_topic</topic>
  </runningDataList>
  <startTimeStamp>1570701361528</startTimeStamp>
  <subscriptionSet>
    <subString>*</subString>
    <subVersion>1570701364301</subVersion>
    <topic>test-mq_topic</topic>
  </subscriptionSet>

```

```

        </subscriptionSet>
        <threadCount>20</threadCount>
        <version>V4_3_6</version>
    </consumerConnectionInfoList>
    <delayTime>0</delayTime>
    <detailInTopicList>
        <delayTime>0</delayTime>
        <lastTimestamp>1570701259403</lastTimestamp>
        <topic>test-mq_topic</topic>
        <totalDiff>0</totalDiff>
    </detailInTopicList>
    <instanceId>MQ_INST_111111111111_DOxxxxxx</instanceId>
    <lastTimestamp>1570701368114</lastTimestamp>
    <online>true</online>
    <rebalanceOK>true</rebalanceOK>
    <subscriptionSame>true</subscriptionSame>
    <totalDiff>0</totalDiff>
</data>
<requestId>10EDC518-10E7-4B34-92FB-171235FA6E38</requestId>
</OnsConsumerStatusResponse>

```

JSON 格式

```

{
  "data": {
    "connectionSet": [
      {
        "bizVersion": "V4_3_6",
        "clientAddr": "30.5.***.*",
        "clientId": "30.5.***.*@97730#-1999745829#-1737591554#729272961762836",
        "language": "JAVA",
        "version": "V4_3_6"
      }
    ],
    "consumeModel": "CLUSTERING",
    "consumeTps": 0,
    "consumerConnectionInfoList": [
      {
        "bizVersion": "V4_3_6",
        "clientId": "30.5.***.*@97730#-1999745829#-1737591554#729272961762836",
        "consumeType": "PUSH",

```

```
"jstack": [
  {
    "thread": "ConsumeMessageThread_1",
    "trackList": [
      "TID: 44 STATE: WAITING",
      "sun.misc.Unsafe.park(Native Method)",
      "java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)",
      "java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2039)",
      "java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)",
      "java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1074)",
      "java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)",
      "java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)",
      "java.lang.Thread.run(Thread.java:748)"
    ]
  },
],
"language": "JAVA",
"lastTimeStamp": 1570701368114,
"runningDataList": [
  {
    "failedCountPerHour": 0,
    "failedTps": 0,
    "okTps": 0,
    "rt": 0,
    "topic": "test-mq_topic"
  },
],
"startTimeStamp": 1570701361528,
"subscriptionSet": [
  {
    "subString": "*",
    "subVersion": 1570701364301,
    "tagsSet": [],
    "topic": "test-mq_topic"
  },
],
"threadCount": 20,
"version": "V4_3_6"
},
],
```

```
"delayTime": 0,
"detailInTopicList": [
  {
    "delayTime": 0,
    "lastTimestamp": 1570701259403,
    "topic": "test-mq_topic",
    "totalDiff": 0
  }
],
"instanceId": "MQ_INST_111111111111_DOxxxxxx",
"lastTimestamp": 1570701368114,
"online": true,
"rebalanceOK": true,
"subscriptionSame": true,
"totalDiff": 0
},
"requestId": "10EDC518-10E7-4B34-92FB-171235FA6E38"
}
```

错误码

访问[错误中心](#)查看更多错误码。

16.3. OnsConsumerGetConnection

使用 OnsConsumerGetConnection 查询指定 Group ID 下当前客户端的连接情况。

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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsConsumerGetConnection	系统规定参数。取值：OnsConsumerGetConnection。
GroupId	String	是	GLD_test_consumer_id	需查询的 Group ID。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需查询的 Group ID 所对应的实例 ID。

返回数据

名称	类型	示例值	描述
Data	Struct		返回参数数据结构。
ConnectionList	Array		指定 Group ID 的连接信息。
ConnectionDo			
ClientAddr	String	30.5.121.**	该消费者实例的地址和端口。
ClientId	String	30.5.121.**@24813#-1999745829#-1737591554#453111174894656	消费者实例的 ID。
Language	String	JAVA	消费者应用的开发语言。
Version	String	V4_3_6	消费端版本。
RequestId	String	DE4140C7-F42D-473D-A5FF-B1E3169245F2	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsConsumerGetConnection
&GroupId=GID_test_consumer_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsConsumerGetConnectionResponse>
  <Data>
    <ConnectionList>
      <ConnectionDo>
        <ClientAddr>30.5.121.**</ClientAddr>
        <Language>JAVA</Language>
        <Version>V4_3_6</Version>
        <ClientId>30.5.121.**@24813#-1999745829#-1737591554#453111174894656</ClientId>
      </ConnectionDo>
    </ConnectionList>
  </Data>
  <RequestId>DE4140C7-F42D-473D-A5FF-B1E3169245F2</RequestId>
</OnsConsumerGetConnectionResponse>
```

JSON 格式

```
{
  "Data": {
    "ConnectionList": {
      "ConnectionDo": [
        {
          "ClientAddr": "30.5.121.**",
          "Language": "JAVA",
          "Version": "V4_3_6",
          "ClientId": "30.5.121.**@24813#-1999745829#-1737591554#453111174894656"
        }
      ]
    }
  },
  "RequestId": "DE4140C7-F42D-473D-A5FF-B1E3169245F2"
}
```

错误码

访问[错误中心](#)查看更多错误码。

16.4. OnsConsumerResetOffset

使用 OnsConsumerResetOffset 重置指定的 Group ID 的消费位点到指定时间戳。

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

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsConsumerResetOffset	系统规定参数。取值：OnsConsumerResetOffset。
GroupId	String	是	GID_test_consumer_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_1111111_111111_DOxxxxxx	需查询的 Group ID 所对应的实例 ID。
Topic	String	是	test-mq-topic	需重置位点的 Topic。
Type	Integer	是	1	消息清理类型。取值说明如下： ● 0：清除所有消息。该 Group ID 在消费该 Topic 下的消息时会跳过当前堆积（未被消费）的所有消息，从这之后发送的最新消息开始消费。 对于程序返回 "reconsumeLater"，即走重试流程的这部分消息来说，清除无效。 ● 1：清理到指定时间。请指定一个时间点，这个时间点之后发送的消息才会被消费。 可选时间范围中的起始和终止时间分别是该 Topic 中储存的最早的和最晚的一条消息的生产时间。不能选择超过可选时间范围的时间点。
ResetTimestamp	Long	否	1591153871000	重置位点到指定的时间戳，仅仅在 Type 为 1 时生效。单位：毫秒。

返回数据

名称	类型	示例值	描述
RequestId	String	D52C68F8-EC5D-4294-BFFF-1A6A25AFBE54	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsConsumerResetOffset
&GroupId=GID_test_consumer_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test-mq-topic
&Type=1
&ResetTimestamp=1591153871000
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsConsumerResetOffsetResponse>
  <RequestId>D52C68F8-EC5D-4294-BFFF-1A6A25AFBE54</RequestId>
</OnsConsumerResetOffsetResponse>
```

JSON 格式

```
{
  "RequestId": "D52C68F8-EC5D-4294-BFFF-1A6A25AFBE54"
}
```

错误码

访问[错误中心](#)查看更多错误码。

16.5. OnsConsumerTimeSpan

使用 OnsConsumerTimeSpan 查询当前 Group ID 订阅的 Topic 可重置的时间范围，即该 Topic 中储存的最早和最晚的一条消息的生产时间。

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。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
----	----	------	-----	----

名称	类型	是否必选	示例值	描述
Action	String	是	OnsConsumerTimeSpan	系统规定参数。取值：OnsConsumerTimeSpan。
GroupId	String	是	GID_test_group_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需查询的消费端 Group ID 所对应的实例 ID。
Topic	String	是	test-mq_topic	需查询的消费端 Group ID 订阅的 Topic。

返回数据

名称	类型	示例值	描述
Data	Struct		查询结果。
ConsumeTimeStamp	Long	1570761026400	当前 Group 消费该 Topic 的最新时间。
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	需查询的消费端 Group ID 所对应的实例 ID。
MaxTimeStamp	Long	1570761026804	整个 Topic 目前最早存储的消息的生产时间。
MinTimeStamp	Long	1570701231122	整个 Topic 目前最新存储的消息的生产时间。
Topic	String	test-mq_topic	需查询的 Topic。
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA1111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsConsumerTimeSpan
&GroupId=GID_test_group_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsConsumerTimeSpanResponse>
  <Data>
    <MinTimeStamp>1570701231122</MinTimeStamp>
    <ConsumeTimeStamp>1570761026400</ConsumeTimeStamp>
    <InstanceId>MQ_INST_111111111111_DOxxxxxx</InstanceId>
    <Topic>test-mq_topic</Topic>
    <MaxTimeStamp>1570761026804</MaxTimeStamp>
  </Data>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsConsumerTimeSpanResponse>
```

JSON 格式

```
{
  "Data":{
    "MinTimeStamp":1570701231122,
    "ConsumeTimeStamp":1570761026400,
    "InstanceId":"MQ_INST_111111111111_DOxxxxxx",
    "Topic":"test-mq_topic",
    "MaxTimeStamp":1570761026804
  },
  "RequestId":"A07E3902-B92E-44A6-B6C5-6AA111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

16.6. OnsMessagePush

使用 OnsMessagePush 向指定的消费者推送消息。

使用场景

OnsMessagePush 接口主要用于消费验证的场景。

使用限制

OnsMessagePush 接口每分钟调用上限为 600 次。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsMessagePush	系统规定参数。取值： OnsMessagePush 。
ClientId	String	是	30.5.121.**@24813#-1999745829#-1737591554#453111174894656	消费者实例的 ID。您可调用 OnsConsumerGetConnection 接口获取该 ID。
GroupId	String	是	GlD_test_group_id	Group ID。详情请参见 名词解释 。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	所需推送的消息所在的实例 ID。
MsgId	String	是	0BC1669963053CF68F733BB70396***	消息的标识，即 Message ID。
Topic	String	是	test-mq_topic	消息推送至的目标 Topic。

返回数据

名称	类型	示例值	描述
RequestId	String	B8EDC90D-F726-4B9E-8BEF-F0DD25EC18D2	公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsMessagePush
&ClientId=30.5.121.**@24813#-1999745829#-1737591554#453111174894656
&GroupId=GlD_test_group_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&MsgId=0BC1669963053CF68F733BB70396****
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsMessagePushResponse>
  <RequestId>B8EDC90D-F726-4B9E-8BEF-F0DD25EC18D2</RequestId>
</OnsMessagePushResponse>
```

JSON 格式

```
{
  "RequestId": "B8EDC90D-F726-4B9E-8BEF-F0DD25EC18D2"
}
```

错误码

访问[错误中心](#)查看更多错误码。

17.消息查询接口

17.1. OnsMessageTrace

使用 OnsMessageTrace 根据 Message ID 来判断目标消息是否曾被消费过。

本接口的实现基于消息队列 RocketMQ 版的内部位点机制，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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsMessageTrace	系统规定参数。取值：OnsMessageTrace。
Instanceld	String	是	MQ_INST_111111111111_DOxxxxxx	需查询的消息所对应的实例 ID。
MsgId	String	是	1E05791C117818B4AAC23B1BB0CE****	需查询消息轨迹的消息 ID，即 Message ID。
Topic	String	是	test-mq_topic	消息的 Topic。

返回数据

名称	类型	示例值	描述
Data	Array		查询结果。
MessageTrack			
ConsumerGroup	String	GI0_test_group_id	订阅该 Topic 的消费者所对应的 Group ID。
Instanceld	String	MQ_INST_111111111111_DOxxxxxx	需查询的消息所对应的实例 ID。

名称	类型	示例值	描述
TrackType	String	CONSUMED	当前状态。取值说明如下： - 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
RequestId	String	EAE5BE23-37A1-4354-94D6-E44AE17E06A7	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsMessageTrace
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&MsgId=1E05791C117818B4AAC23B1BB0CE****
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsMessageTraceResponse>
  <Data>
    <MessageTrack>
      <InstanceId>MQ_INST_111111111111_DOxxxxxx</InstanceId>
      <ConsumerGroup>GID_test_group_id</ConsumerGroup>
      <TrackType>CONSUMED</TrackType>
    </MessageTrack>
  </Data>
  <RequestId>EAE5BE23-37A1-4354-94D6-E44AE17E06A7</RequestId>
</OnsMessageTraceResponse>
```

JSON 格式


```
{
  "Data": {
    "MessageTrack": [
      {
        "Instanceld": "MQ_INST_111111111111_DOxxxxxx",
        "ConsumerGroup": "GID_test_group_id",
        "TrackType": "CONSUMED"
      }
    ]
  },
  "RequestId": "EAE5BE23-37A1-4354-94D6-E44AE17E06A7"
}
```

错误码

访问[错误中心](#)查看更多错误码。

17.2. OnsMessageGetByMsgId

调用 OnsMessageGetByMsgId 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。

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。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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsMessageGetByMsgId	系统规定参数。取值：OnsMessageGetByMsgId。
Instanceld	String	是	MQ_INST_111111111111_DOxxxxxx	需查询消息所对应的实例 ID。
MsgId	String	是	1E0578FE110F18B4AAC235C05F2***	需查询的消息的 ID，即 Message ID。
Topic	String	是	test-mq_topic	需查询的消息的 Topic。

 说明 更多参数说明请参见[公共参数](#)和[调用方式](#)。

返回数据

名称	类型	示例值	描述
Data	Struct		查询结果。
BodyCRC	Integer	914112295	消息体 CRC 校验值。
BornHost	String	42.120.**.*:64646	生成该消息的客户端实例。
BornTimestamp	Long	1570761026630	生成时间戳。
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
MsgId	String	1E0578FE110F18B4AAC235C0C8460BA2	消息 ID，即 Message ID。
PropertyList	Array		消息属性列表。
MessageProperty			
Name	String	TAGS	消息属性。取值说明如下： - TRACE_ON：是否有消息轨迹 - KEYS：消息的 Key 属性 - TAGS：消息的 Tag 属性 - INSTANCE_ID：消息所在的实例的 ID 更多概念信息请参见 名词解释 。
Value	String	TagA	消息属性的值。
ReconsumeTimes	Integer	1	消息重试消费的次数。
StoreHost	String	11.220.***.***:10911	存储该消息的服务器实例。
StoreSize	Integer	407	消息大小。

名称	类型	示例值	描述
StoreTimestamp	Long	1570761026708	被服务端存储的时间戳。
Topic	String	test-mq_topic	消息的 Topic。
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsMessageGetByMsgId
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&MsgId=1E0578FE110F18B4AAC235C05F2*****
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsMessageGetByMsgIdResponse>
  <Data>
    <MsgId>1E0578FE110F18B4AAC235C0C8460BA2</MsgId>
    <Instanceld>MQ_INST_111111111111_DOxxxxxx</Instanceld>
    <ReconsumeTimes>1</ReconsumeTimes>
    <BornHost>42.120.**.**:64646</BornHost>
    <BodyCRC>914112295</BodyCRC>
    <StoreHost>11.220.**.**:10911</StoreHost>
    <StoreTimestamp>1570761026708</StoreTimestamp>
    <BornTimestamp>1570761026630</BornTimestamp>
    <Topic>test-mq_topic</Topic>
    <StoreSize>407</StoreSize>
    <PropertyList>
      <MessageProperty>
        <Name>TAGS</Name>
        <Value>TagA</Value>
      </MessageProperty>
    </PropertyList>
  </Data>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsMessageGetByMsgIdResponse>
```

JSON 格式

```
{
  "Data": {
    "MsgId": "1E0578FE110F18B4AAC235C0C8460BA2",
    "InstancId": "MQ_INST_111111111111_DOxxxxxx",
    "ReconsumeTimes": 1,
    "BornHost": "42.120.**.**:64646",
    "BodyCRC": 914112295,
    "StoreHost": "11.220.**.**:10911",
    "StoreTimestamp": 1570761026708,
    "BornTimestamp": 1570761026630,
    "Topic": "test-mq_topic",
    "StoreSize": 407,
    "PropertyList": {
      "MessageProperty": [
        {
          "Name": "TAGS",
          "Value": "TagA"
        }
      ]
    }
  },
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA1111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

17.3. OnsMessageGetByKey

调用 OnsMessageGetByKey 根据 Topic 和 Message Key 模糊查询消息信息列表。

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。OnsMessageGetByMsgId 接口的详细信息，请参见 [OnsMessageGetByMsgId](#)。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsMessageGetByKey	系统规定参数。取值：OnsMessageGetByKey。
Instanceld	String	是	MQ_INST_111111111111_DOxxxxxx	需查询的消息所对应的实例 ID。
Key	String	是	messageKey1	需查询的消息的 Message Key。
Topic	String	是	test-mq_topic	需查询的消息的 Topic。

 说明 更多参数说明请参见[公共参数](#)和[调用方式](#)。

返回数据

名称	类型	示例值	描述
Data	Array		查询结果。
OnsRestMessageDo			
BodyCRC	Integer	914112295	消息体 CRC 校验值。
BornHost	String	42.120.***.***:59270	生成该消息的客户端实例。
BornTimestamp	Long	1570760999721	生成时间戳。
Instanceld	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
MsgId	String	1E0578FE110F18B4AAC235C05F2*****	消息 ID，即 Message ID。
PropertyList	Array		消息属性列表。
MessageProperty			

名称	类型	示例值	描述
Name	String	TAGS	消息属性。取值说明如下： - TRACE_ON：是否有消息轨迹 - KEYS：消息的 Key 属性 - TAGS：消息的 Tag 属性 - INSTANCE_ID：消息所在的实例的 ID 更多概念信息请参见 名词解释 。
Value	String	TagA	消息属性的取值。
ReconsumeTimes	Integer	1	消息重试消费的次数。
StoreHost	String	11.193.***.***:10911	存储该消息的服务器实例。
StoreSize	Integer	406	消息大小。
StoreTimestamp	Long	1570760999811	被 Broker 存储的时间戳。
Topic	String	test-mq_topic	消息的 Topic。
RequestId	String	5DC2A47E-2B31-4722-96C8-FA59C996F0CA	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsMessageGetByKey
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Key=messageKey1
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsMessageGetByKeyResponse>
  <Data>
    <MsgId>1E0578FE110F18B4AAC235C05F2*****</MsgId>
    <Instanceld>MQ_INST_111111111111_DOxxxxxx</Instanceld>
    <ReconsumeTimes>1</ReconsumeTimes>
    <BornHost>42.120.**.*:64646</BornHost>
    <BodyCRC>914112295</BodyCRC>
    <StoreHost>11.220.**.*:10911</StoreHost>
    <StoreTimestamp>1570761026708</StoreTimestamp>
    <BornTimestamp>1570761026630</BornTimestamp>
    <Topic>test-mq_topic</Topic>
    <StoreSize>407</StoreSize>
    <PropertyList>
      <MessageProperty>
        <Name>TAGS</Name>
        <Value>TagA</Value>
      </MessageProperty>
    </PropertyList>
  </Data>
  <RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsMessageGetByKeyResponse>
```

JSON 格式


```
{
  "Data": {
    "MsgId": "1E0578FE110F18B4AAC235C05F2*****",
    "InstancId": "MQ_INST_111111111111_DOxxxxxx",
    "ReconsumeTimes": 1,
    "BornHost": "42.120.**.**:64646",
    "BodyCRC": 914112295,
    "StoreHost": "11.220.***.***:10911",
    "StoreTimestamp": 1570761026708,
    "BornTimestamp": 1570761026630,
    "Topic": "test-mq_topic",
    "StoreSize": 407,
    "PropertyList": {
      "MessageProperty": [
        {
          "Name": "TAGS",
          "Value": "TagA"
        }
      ]
    }
  },
  "RequestId": "A07E3902-B92E-44A6-B6C5-6AA1111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

17.4. OnsMessagePageQueryByTopic

调用 OnsMessagePageQueryByTopic 分页查询指定时间段内指定 Topic 内存在的所有消息。

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

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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsMessagePageQueryByTopic	系统规定参数。取值：OnsMessagePageQueryByTopic。
BeginTime	Long	是	1570723200000	查询范围的起始时间戳。
CurrentPage	Integer	是	2	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。
EndTime	Long	是	1570809600000	查询范围的终止时间戳。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需查询的消息所对应的实例 ID。
Topic	String	是	test-mq_topic	需查询的消息的 Topic 名称。
TaskId	String	否	0BC1310300002A9F000021E4D7A48346	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。
PageSize	Integer	否	5	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。

 说明 更多参数说明请参见公共参数和调用方式。

返回数据

名称	类型	示例值	描述
MsgFoundDo	Struct		查询结果。
CurrentPage	Long	1	当前位于第几页。
MaxPageCount	Long	400	查询结果最大页数。

名称	类型	示例值	描述
MsgFoundList	Array		当前页的消息集合，该类型与 OnsMessageGetByMsgId 的返回结果一致。
OnsRestMessageDo			
BodyCRC	Integer	914112295	消息体 CRC 校验值。
BornHost	String	42.120.***.***:59270	生成该消息的客户端实例。
BornTimestamp	Long	1570760999721	生成时间戳。
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
MsgId	String	1E0578FE110F18B4AAC235C05F2*****	消息 ID，即 Message ID。
PropertyList	Array		消息属性查询。
MessageProperty			
Name	String	TAGS	消息属性。取值说明如下： - TRACE_ON：是否有消息轨迹 - KEYS：消息的 Key 属性 - TAGS：消息的 Tag 属性 - INSTANCE_ID：消息所在的实例的 ID 更多概念信息请参见 名词解释 。
Value	String	TagA	消息属性的取值。
ReconsumeTimes	Integer	1	消息重试消费的次数。
StoreHost	String	11.193.***.***:10911	存储该消息的服务器实例。
StoreSize	Integer	406	消息大小。
StoreTimestamp	Long	1570760999811	被服务端存储的时间戳。

名称	类型	示例值	描述
Topic	String	test-mq_topic	消息的 Topic。
TaskId	String	0BC1310300002A9F000021E4D7A48346	第一次新建查询时，返回的查询任务 ID，用于后续取消息。
RequestId	String	5DC2A47E-2B31-4722-96C8-FA59C9*****	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsMessagePageQueryByTopic
&BeginTime=1570723200000
&CurrentPage=2
&EndTime=1570809600000
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test-mq_topic
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsMessagePageQueryByTopicResponse>
  <MsgFoundDo>
    <MaxPageCount>400</MaxPageCount>
    <MsgFoundList>
      <OnsRestMessageDo>
        <MsgId>1E0578FE110F18B4AAC235C05F2*****</MsgId>
        <Instanceld>MQ_INST_111111111111_DOxxxxxx</Instanceld>
        <ReconsumeTimes>1</ReconsumeTimes>
        <BornHost>42.120.***.***:59270</BornHost>
        <StoreTimestamp>157076099811</StoreTimestamp>
        <StoreHost>11.193.***.***:10911</StoreHost>
        <BodyCRC>914112295</BodyCRC>
        <Topic>test-mq_topic</Topic>
        <BornTimestamp>1570760999721</BornTimestamp>
        <StoreSize>406</StoreSize>
        <PropertyList>
          <MessageProperty>
            <Name>TAGS</Name>
            <Value>TagA</Value>
          </MessageProperty>
        </PropertyList>
      </OnsRestMessageDo>
    </MsgFoundList>
    <TaskId>0BC1310300002A9F000021E4D7A48346</TaskId>
    <CurrentPage>1</CurrentPage>
  </MsgFoundDo>
  <RequestId>5DC2A47E-2B31-4722-96C8-FA59C9*****</RequestId>
</OnsMessagePageQueryByTopicResponse>
```

JSON 格式

```
{
  "MsgFoundDo": {
    "MaxPageCount": 400,
    "MsgFoundList": {
      "OnsRestMessageDo": [
        {
          "MsgId": "1E0578FE110F18B4AAC235C05F2*****",
          "ReconsumeTimes": 1,
          "BornHost": "42.120.***.***:59270",
          "StoreTimestamp": 1570760999811,
          "StoreHost": "11.193.***.***:10911",
          "BodyCRC": 914112295,
          "Topic": "test-mq_topic",
          "BornTimestamp": 1570760999721,
          "StoreSize": 406,
          "PropertyList": {
            "MessageProperty": [
              {
                "Name": "TAGS",
                "Value": "TagA"
              }
            ]
          }
        }
      ]
    }
  },
  "TaskId": "0BC1310300002A9F000021E4D7A48346",
  "CurrentPage": 1
},
"RequestId": "5DC2A47E-2B31-4722-96C8-FA59C9*****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

18.发布订阅统计接口

18.1. OnsTrendTopicInputTps

使用 OnsTrendTopicInputTps 查询一段时间内指定的 Topic 的消息写入报表数据。

使用说明

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，可以根据需要选择查询指定时间段内 Topic 的写入量或者 TPS 曲线数据。

 **说明** 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。

使用限制

OnsTrendTopicInputTps 调用频次请勿超过 1 次/分钟。如有监控需求，请使用监控报警功能。详情请参见[监控报警](#)。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTrendTopicInputTps	系统规定参数。取值：OnsTrendTopicInputTps。
BeginTime	Long	是	1570852800000	查询区间的起始时间戳，单位：毫秒。
EndTime	Long	是	1570868400000	查询区间的终止时间戳，单位：毫秒。
Instanceld	String	是	MQ_INST_111111111111_DOxxxxxx	指定的 Topic 所对应的实例 ID。
Topic	String	是	test	需查询的 Topic 名称。
Type	Integer	是	0	查询的类型。取值说明如下： 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。

名称	类型	是否必选	示例值	描述
Period	Long	否	10	采样周期，单位分钟，取值可为 1、5、10。

返回数据

名称	类型	示例值	描述
Data	Struct		数据集合。
Records	Array		采集点信息。
StatsDataDo			
X	Long	1570852800000	横轴，时间戳，单位：毫秒。
Y	Float	0	纵轴，数据（TPS 或者总量）。
Title	String	MQ_INST_111111111111_DOxxxxxx%test 接收 总量 趋势图	Table 的名称。
XUnit	String	time	时间戳，单位：毫秒。
YUnit	String	msg	纵轴单位。
RequestId	String	E213AD8A-0730-4B3D-A35A-340DA47DBB2F	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTrendTopicInputTps
&BeginTime=1570852800000
&EndTime=1570868400000
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test
&Type=0
&<公共请求参数>
```


正常返回示例

XML 格式

```
<OnsTrendTopicInputTpsResponse>
  <Data>
    <Records>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570852800000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570853400000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570854000000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570854600000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570860600000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570861200000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570861800000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570862400000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570863000000</X>
      </StatsDataDo>
    </Records>
  </Data>
</OnsTrendTopicInputTpsResponse>
```

```
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570863600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570864200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>1</Y>
  <X>1570864800000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570865400000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570866000000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570866600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570867200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570867800000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570868400000</X>
</StatsDataDo>
</Records>
<XUnit>time</XUnit>
<YUnit>msg</YUnit>
<Title>MQ INST 111111111111 DQxxxxx%test-mq topic 接收 总量 折线图</Title>
```

```
<Data></Data>  
<RequestId>E213AD8A-0730-4B3D-A35A-340DA47DBB2F</RequestId>  
</OnsTrendTopicInputTpsResponse>
```

JSON 格式

```
{
  "Data": {
    "Records": {
      "StatsDataDo": [
        {
          "Y": 0,
          "X": 1570852800000
        },
        {
          "Y": 0,
          "X": 1570853400000
        },
        {
          "Y": 0,
          "X": 1570854000000
        },
        {
          "Y": 0,
          "X": 1570854600000
        },
        {
          "Y": 0,
          "X": 1570860600000
        },
        {
          "Y": 0,
          "X": 1570861200000
        },
        {
          "Y": 0,
          "X": 1570861800000
        },

```

```
"X": 1570862400000
},
{
  "Y": 0,
  "X": 1570863000000
},
{
  "Y": 0,
  "X": 1570863600000
},
{
  "Y": 0,
  "X": 1570864200000
},
{
  "Y": 1,
  "X": 1570864800000
},
{
  "Y": 0,
  "X": 1570865400000
},
{
  "Y": 0,
  "X": 1570866000000
},
{
  "Y": 0,
  "X": 1570866600000
},
{
  "Y": 0,
  "X": 1570867200000
},
{
  "Y": 0,
  "X": 1570867800000
},
{
  "Y": 0,
  "X": 1570868400000
```

```
    }
  ]
},
  "XUnit": "time",
  "YUnit": "msg",
  "Title": "MQ_INST_111111111111_DOxxxxxx%test 接收 总量 趋势图"
},
  "RequestId": "E213AD8A-0730-4B3D-A35A-340DA47DBB2F"
}
```

错误码

访问[错误中心](#)查看更多错误码。

18.2. OnsTrendGroupOutputTps

使用 OnsTrendGroupOutputTps 查询指定的 Group ID 在一段时间内消费消息的统计信息。

使用说明

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

 **说明** 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。

使用限制

OnsTrendGroupOutputTps 调用频次请勿超过 1 次/分钟。如有监控需求，请使用监控报警功能。详情请参见[监控报警](#)。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTrendGroupOutputTps	系统规定参数。取值：OnsTrendGroupOutputTps。
BeginTime	Long	是	1570852800000	查询区间的起始时间戳，单位：毫秒。

名称	类型	是否必选	示例值	描述
EndTime	Long	是	1570868400000	查询区间的终止时间戳，单位：毫秒。
GroupId	String	是	GID_test	需要查询的 Group ID。
InstanceId	String	是	MQ_INST_1111111 111111_DOxxxxxx	需统计数据的 Group ID 所对应的实例 ID。
Topic	String	是	test	需要查询的 Topic 名称。
Type	Integer	是	0	查询的类型。取值说明如下： 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。
Period	Long	否	10	采样周期，单位分钟，取值可为 1、5、10。

返回数据

名称	类型	示例值	描述
Data	Struct		数据集合。
Records	Array		采集点信息。
StatsDataDo			
X	Long	1570867800000	横轴，时间戳，单位：毫秒。
Y	Float	0	纵轴，数据（TPS 或者总量）。
Title	String	MQ_INST_111111111 111_DOxxxxxx%test @MQ_INST_1111111 11111_DOxxxxxx%GI D_test 投递 总量 趋 势图	Table 的名称。
XUnit	String	time	时间戳，单位：毫秒。
YUnit	String	msg	数量。

名称	类型	示例值	描述
RequestId	String	CE57AEDC-8FD2-43ED-8E3B-1F8780779FE2	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTrendGroupOutputTps
&BeginTime=1570852800000
&EndTime=1570868400000
&GroupId=GID_test
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&Topic=test
&Type=0
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTrendGroupOutputTpsResponse>
  <Data>
    <Records>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570852800000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570853400000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570854000000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
        <X>1570854600000</X>
      </StatsDataDo>
      <StatsDataDo>
        <Y>0</Y>
      </StatsDataDo>
```

```
<X>1570855200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570860600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570861200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570861800000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570862400000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570863000000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570863600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570864200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>1</Y>
  <X>1570864800000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570865400000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570866000000</X>
```



```

</X>1570866600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570866600000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570867200000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570867800000</X>
</StatsDataDo>
<StatsDataDo>
  <Y>0</Y>
  <X>1570868400000</X>
</StatsDataDo>
</Records>
<XUnit>time</XUnit>
<YUnit>数量</YUnit>
<Title>MQ_INST_111111111111_DOxxxxxx%test@MQ_INST_111111111111_DOxxxxxx%GID_test 投递 总量
趋势图</Title>
</Data>
<RequestId>CE57AEDC-8FD2-43ED-8E3B-1F8780779FE2</RequestId>
</OnsTrendGroupOutputTpsResponse>

```

JSON 格式

```

{
  "Data": {
    "Records": {
      "StatsDataDo": [
        {
          "Y": 0,
          "X": 1570852800000
        },
        {
          "Y": 0,
          "X": 1570853400000
        },
        .
      ]
    }
  }
}

```

```
{
  "Y": 0,
  "X": 1570854000000
},
{
  "Y": 0,
  "X": 1570854600000
},
{
  "Y": 0,
  "X": 1570855200000
},
{
  "Y": 0,
  "X": 1570860600000
},
{
  "Y": 0,
  "X": 1570861200000
},
{
  "Y": 0,
  "X": 1570861800000
},
{
  "Y": 0,
  "X": 1570862400000
},
{
  "Y": 0,
  "X": 1570863000000
},
{
  "Y": 0,
  "X": 1570863600000
},
{
  "Y": 0,
  "X": 1570864200000
},
{
```

```
"Y": 1,
  "X": 1570864800000
},
{
  "Y": 0,
  "X": 1570865400000
},
{
  "Y": 0,
  "X": 1570866000000
},
{
  "Y": 0,
  "X": 1570866600000
},
{
  "Y": 0,
  "X": 1570867200000
},
{
  "Y": 0,
  "X": 1570867800000
},
{
  "Y": 0,
  "X": 1570868400000
}
]
},
  "XUnit": "time",
  "YUnit": "数量",
  "Title": "MQ_INST_111111111111_DOxxxxxx%test@MQ_INST_111111111111_DOxxxxxx%GID_test 投递 总量 趋势图"
},
  "RequestId": "CE57AEDC-8FD2-43ED-8E3B-1F8780779FE2"
}
```

错误码

访问[错误中心](#)查看更多错误码。

19.消息轨迹接口

19.1. OnsTraceQueryByMsgId

使用 OnsTraceQueryByMsgId 通过传入 Topic 名称和 Message ID 创建轨迹查询任务，得到该查询任务的 ID。

业务方记录了某消息的 Message ID，并需要根据此 Message ID 查询该消息的投递轨迹信息，即可使用本接口创建查询任务。该接口返回任务 ID 后，再根据 **OnsTraceGetResult** 接口传入任务 ID 即可获取查询结果。**OnsTraceGetResult** 接口的详细信息，请参见 **OnsTraceGetResult**。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTraceQueryByMsgId	系统规定参数。取值：OnsTraceQueryByMsgId。
BeginTime	Long	是	1570852800	查询的时间范围的起始时间点。
EndTime	Long	是	1570968000	查询的时间范围的结束时间点。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需查询的消息的 Topic 所对应的实例 ID。
MsgId	String	是	1E05791C117818B4AAC23B1BB0CE****	需查询的消息的 ID，即 Message ID。
Topic	String	是	test	需查询的消息的 Topic。

返回数据

名称	类型	示例值	描述
QueryId	String	272967562652883649157096685****	该查询任务的 ID，可根据该任务 ID 再调用 OnsTraceGetResult 接口获取详细结果。
RequestId	String	B93332A3-160D-404F-880F-1F8736D1039F	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTraceQueryByMsgId
&BeginTime=1570852800
&EndTime=1570968000
&InstanceId=MQ_INST_11111111111_DOxxxxxx
&MsgId=1E05791C117818B4AAC23B1BB0CE****
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTraceQueryByMsgIdResponse>
  <RequestId>B93332A3-160D-404F-880F-1F8736D1039F</RequestId>
  <QueryId>272967562652883649157096685****</QueryId>
</OnsTraceQueryByMsgIdResponse>
```

JSON 格式

```
{
  "RequestId": "B93332A3-160D-404F-880F-1F8736D1039F",
  "QueryId": "272967562652883649157096685****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

19.2. OnsTraceGetResult

使用 OnsTraceGetResult 通过传入轨迹查询任务的 ID 获取之前的轨迹查询结果。

业务方已经根据 Message ID 或者 Message Key 创建了轨迹查询任务，并得到了查询任务的 ID。您可调用本接口根据该任务 ID 获取对应的轨迹结果的明细。

因为消息轨迹查询任务耗时较多，创建任务后立即根据该接口获取结果可能为空，若遇到此情况，建议等待一段时间再重试。

调试

您可以在 OpenAPI Explorer 中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer 可以自动生成 SDK 代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTraceGetResult	系统规定参数。取值：OnsTraceGetResult。
QueryId	String	是	272967562652883649157096685**	需要提取轨迹查询结果的任务 ID。

返回数据

名称	类型	示例值	描述
TraceData	Struct		该查询任务的结果明细。
CreateTime	Long	1570966857000	该查询任务的创建时间。
InstanceId	String	MQ_INST_11111111111_DOxxxxxx	实例 ID。
MsgId	String	1E05791C117818B4AAC23B1BB0CE****	该查询任务对应的 Message ID。
MsgKey	String	ORDERID_100	该查询任务对应的 Message Key。
QueryId	String	272967562652883649157096685****	查询任务 ID。
Status	String	finish	该查询任务的结果。取值说明如下： finish：查询完成 working：查询中 removed：任务已删除
Topic	String	test	该查询任务的 Topic。
TraceList	Array		该查询任务的匹配轨迹列表。
TraceMapDo			
BornHost	String	30.5.**.**	消息发送方的客户端地址。
CostTime	Integer	24	发送耗时，单位毫秒。

名称	类型	示例值	描述
MsgId	String	0BC1F01800002A9F00000531246****	消息的 ID，即 Message ID。
MsgKey	String	ORDERID_100	消息的 Key，即 Message Key。
PubGroupName	String	GIID_test	发送方客户端配置的 Group ID。
PubTime	Long	1570850870478	消息发送时间。
Status	String	SEND_SUCCESS	发送状态。取值说明如下： - SEND_SUCCESS：发送成功 - SEND_FAILED：发送失败 - SEND_ROLLBACK：事务消息回滚 - SEND_UNKNOWN：事务消息未提交 - SEND_DELAY：定时（延时）消息定时中
SubList	Array		消息的消费轨迹列表。
SubMapDo			
ClientList	Array		该 Group ID 客户端消费记录的明细列表。
SubClientInfoDo			
ClientHost	String	30.5.**.**	消费客户端的客户端地址。
CostTime	Integer	43	本次消费耗时，单位毫秒。
ReconsumeTimes	Integer	1	本次消费的投递轮次。
Status	String	CONSUME_SUCCESS	消费状态。取值说明如下： - CONSUME_FAILED：消费失败 - CONSUME_SUCCESS：消费成功 - CONSUME_NOT_RETURN：消费未返回结果 - SEND_UNKNOWN：事务消息未提交 - SEND_DELAY：定时（延时）消息定时中

名称	类型	示例值	描述
SubGroupName	String	GlD_test	该客户端所属的 Group ID。
SubTime	Long	1570851590511	消费开始时间戳。
FailCount	Integer	0	该 Group ID 消费失败次数统计。
SubGroupName	String	GlD_test	消费方 Group ID。
SuccessCount	Integer	1	该 Group ID 消费成功次数统计。
Tag	String	TagA	消息的 Tag，即 Message Tag。
Topic	String	test	消息的 Topic。
UpdateTime	Long	1570966877000	该查询任务的最后更新时间。
UserId	String	27296756265288****	查询用户 ID。
RequestId	String	84EE24D2-851F-40D6-B99E-4D6AB9098BC4	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTraceGetResult
&QueryId=272967562652883649157096685****
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTraceGetResultResponse>
  <TraceData>
    <MsgId>1E05791C117818B4AAC23B1BB0CE****</MsgId>
    <Status>finish</Status>
    <TraceList>
      <TraceManDo>
```



```

    <BornHost>30.5.**.**</BornHost>
    <MsgId>0BC1F01800002A9F000000531246****</MsgId>
    <Status>SEND_SUCCESS</Status>
    <CostTime>24</CostTime>
    <PubTime>1570850870478</PubTime>
    <Tag>TagA</Tag>
    <Topic>test</Topic>
    <MsgKey>ORDERID_100</MsgKey>
    <PubGroupName>GID_test</PubGroupName>
    <SubList>
      <SubMapDo>
        <ClientList>
          <SubClientInfoDo>
            <Status>CONSUME_SUCCESS</Status>
            <CostTime>43</CostTime>
            <ClientHost>30.5.**.**</ClientHost>
            <ReconsumeTimes>1</ReconsumeTimes>
            <SubTime>1570851590511</SubTime>
            <SubGroupName>GID_test</SubGroupName>
          </SubClientInfoDo>
        </ClientList>
        <SuccessCount>1</SuccessCount>
        <FailCount>0</FailCount>
        <SubGroupName>GID_test</SubGroupName>
      </SubMapDo>
    </SubList>
  </TraceMapDo>
</TraceList>
<CreateTime>1570966857000</CreateTime>
<Instanceld>MQ_INST_1111111111_DOxxxxxx</Instanceld>
<Topic>test</Topic>
<UpdateTime>1570966877000</UpdateTime>
<QueryId>272967562652883649157096685****</QueryId>
<UserId>27296756265288****</UserId>
</TraceData>
<RequestId>84EE24D2-851F-40D6-B99E-4D6AB9098BC4</RequestId>
</OnsTraceGetResultResponse>

```

JSON 格式

r

```
{
  "TraceData": {
    "MsgId": "1E05791C117818B4AAC23B1BB0CE****",
    "Status": "finish",
    "TraceList": {
      "TraceMapDo": [
        {
          "BornHost": "30.5.**.**",
          "MsgId": "0BC1F01800002A9F000000531246****",
          "Status": "SEND_SUCCESS",
          "CostTime": 24,
          "PubTime": 1570850870478,
          "Tag": "TagA",
          "Topic": "test",
          "MsgKey": "ORDERID_100",
          "PubGroupName": "GID_test",
          "SubList": {
            "SubMapDo": [
              {
                "ClientList": {
                  "SubClientInfoDo": [
                    {
                      "Status": "CONSUME_SUCCESS",
                      "CostTime": 43,
                      "ClientHost": "30.5.**.**",
                      "ReconsumeTimes": 1,
                      "SubTime": 1570851590511,
                      "SubGroupName": "GID_test"
                    }
                  ]
                }
              }
            ],
            "SuccessCount": 1,
            "FailCount": 0,
            "SubGroupName": "GID_test"
          }
        }
      ],
      "CreateTime": 1570966857000,
```

```
"InstanceId": "MQ_INST_111111111111_DOxxxxxx",
"Topic": "test",
"UpdateTime": 1570966877000,
"QueryId": "272967562652883649157096685****",
"UserId": "27296756265288****"
},
"RequestId": "84EE24D2-851F-40D6-B99E-4D6AB9098BC4"
}
```

错误码

访问[错误中心](#)查看更多错误码。

19.3. OnsTraceQueryByMsgKey

使用 OnsTraceQueryByMsgKey 通过传入 Topic 名称和 Message Key 创建轨迹查询任务，得到该查询任务的 ID。

业务方记录了 Message Key，并需要根据此 Message Key 查询该消息的投递轨迹信息，即可使用 **OnsTraceQueryByMsgKey** 接口创建查询任务。该接口返回任务 ID 后，再根据 **OnsTraceGetResult** 接口传入任务 ID 即可获取查询结果。**OnsTraceGetResult** 接口的详细信息，请参见 [OnsTraceGetResult](#)。

调试

您可以在[OpenAPI Explorer](#)中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsTraceQueryByMsgKey	系统规定参数。取值：OnsTraceQueryByMsgKey。
BeginTime	Long	是	1570852800	查询的时间范围的起始时间点。
EndTime	Long	是	1570968000	查询的时间范围的结束时间点。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需查询的消息的 Topic 所对应的实例 ID。
MsgKey	String	是	ORDERID_100	需查询的消息的 Message Key。
Topic	String	是	test	需查询的消息的 Topic。

返回数据

名称	类型	示例值	描述
QueryId	String	272967562652883649157096685****	该查询任务的 ID，根据该任务 ID 再调用 OnsTraceGetResult 接口获取详细结果。
RequestId	String	F8654231-122A-4DBD-801F-38E3553851A8	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://[Endpoint]/?Action=OnsTraceQueryByMsgKey
&BeginTime=1570852800
&EndTime=1570968000
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&MsgKey=ORDERID_100
&Topic=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsTraceQueryByMsgKeyResponse>
  <RequestId>F8654231-122A-4DBD-801F-38E3553851A8</RequestId>
  <QueryId>272967562652883649157096685****</QueryId>
</OnsTraceQueryByMsgKeyResponse>
```

JSON 格式

```
{
  "RequestId": "F8654231-122A-4DBD-801F-38E3553851A8",
  "QueryId": "272967562652883649157096685****"
}
```

错误码

访问[错误中心](#)查看更多错误码。

20.死信队列接口

20.1. OnsDLQMessageGetById

调用 OnsDLQMessageGetById 通过传入 Message ID 查询指定的死信消息。

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。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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsDLQMessageGetById	系统规定参数。取值：OnsDLQMessageGetById。
GroupId	String	是	GID_test_group_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_111111111111_DOxxxxxx	需查询消息所对应的实例 ID。
MsgId	String	是	0BC16699165C03B925DB8A404E2D****	需查询的消息的 ID，即 Message ID。

 说明 更多参数说明请参见公共参数和调用方式。

返回数据

名称	类型	示例值	描述
Data	Struct		查询结果。
BodyCRC	Integer	914112295	消息体 CRC 校验值。
BornHost	String	42.120.**.**:64646	生成该消息的生产者实例。
BornTimestamp	Long	1570761026630	该条消息被服务端存储的时间戳。

名称	类型	示例值	描述
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
MsgId	String	0BC16699165C03B925DB8A404E2D****	消息 ID。
PropertyList	Array		消息属性列表。
MessageProperty			
Name	String	TAGS	消息属性。取值说明如下： • TRACE_ON：是否有消息轨迹 • KEYS：消息的 Key 属性 • TAGS：消息的 Tag 属性 • INSTANCE_ID：消息所在的实例的 ID 更多概念信息请参见 名词解释 。
Value	String	TagA	消息属性的取值。
ReconsumeTimes	Integer	1	消息重试消费的次数。
StoreHost	String	11.220.***.***:10911	存储该消息的服务器实例。
StoreSize	Integer	407	消息大小。
StoreTimestamp	Long	1570761026708	该条消息被服务端存储的时间戳。
Topic	String	test-mq_topic	消息的 Topic。
RequestId	String	A07E3902-B92E-44A6-B6C5-6AA111111799	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```

http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsDLQMessageGetById
&GroupId=GID_test_group_id
&Instanceld=MQ_INST_11111111111_DOxxxxxx
&MsgId=0BC16699165C03B925DB8A404E2D****
&<公共请求参数>

```

正常返回示例

XML 格式

```

<OnsDLQMessageGetByIdResponse>
  <Data>
    <MsgId>1E0578FE110F18B4AAC235C0C8460BA2</MsgId>
    <Instanceld>MQ_INST_11111111111_DOxxxxxx</Instanceld>
    <ReconsumeTimes>1</ReconsumeTimes>
    <BornHost>42.120.**.*:64646</BornHost>
    <BodyCRC>914112295</BodyCRC>
    <StoreHost>11.220.***.***:10911</StoreHost>
    <StoreTimestamp>1570761026708</StoreTimestamp>
    <BornTimestamp>1570761026630</BornTimestamp>
    <Topic>test-mq_topic</Topic>
    <StoreSize>407</StoreSize>
    <PropertyList>
      <MessageProperty>
        <Name>TRACE_ON</Name>
        <Value>true</Value>
      </MessageProperty>
      <MessageProperty>
        <Name>RETRY_TOPIC</Name>
        <Value>MQ_INST_11111111111_DOxxxxxx%test-mq_topic</Value>
      </MessageProperty>
      <MessageProperty>
        <Name>KEYS</Name>
        <Value>ORDERID_993</Value>
      </MessageProperty>
      <MessageProperty>
        <Name>UNIQ_KEY</Name>
        <Value>1E0578FE110F18B4AAC235C0C84****</Value>
      </MessageProperty>
      <MessageProperty>
        <Name>WAIT</Name>

```

```

        <Value>false</Value>
    </MessageProperty>
    <MessageProperty>
        <Name>TAGS</Name>
        <Value>TagA</Value>
    </MessageProperty>
    <MessageProperty>
        <Name>INSTANCE_ID</Name>
        <Value>MQ_INST_111111111111_DOxxxxxx</Value>
    </MessageProperty>
</PropertyList>
</Data>
<RequestId>A07E3902-B92E-44A6-B6C5-6AA111111799</RequestId>
</OnsDLQMessageGetByIdResponse>

```

JSON 格式

```

{
  "Data": {
    "MsgId": "0BC16699165C03B925DB8A404E2D****",
    "InstancId": "MQ_INST_111111111111_DOxxxxxx",
    "ReconsumeTimes": 1,
    "BornHost": "42.120.**.**:64646",
    "BodyCRC": 914112295,
    "StoreHost": "11.220.**.**:10911",
    "StoreTimestamp": 1570761026708,
    "BornTimestamp": 1570761026630,
    "Topic": "test-mq_topic",
    "StoreSize": 407,
    "PropertyList": {
      "MessageProperty": [
        {
          "Name": "TRACE_ON",
          "Value": "true"
        },
        {
          "Name": "RETRY_TOPIC",
          "Value": "MQ_INST_111111111111_DOxxxxxx%test-mq_topic"
        },
        {
          "Name": "KEYS",

```



```
{
  "Value": "ORDERID_993"
},
{
  "Name": "UNIQ_KEY",
  "Value": "1E0578FE110F18B4AAC235C0C84*****"
},
{
  "Name": "WAIT",
  "Value": "false"
},
{
  "Name": "TAGS",
  "Value": "TagA"
},
{
  "Name": "INSTANCE_ID",
  "Value": "MQ_INST_111111111111_DOxxxxxx"
}
]
}
},
"RequestId": "A07E3902-B92E-44A6-B6C5-6AA111111799"
}
```

错误码

访问[错误中心](#)查看更多错误码。

20.2.

OnsDLQMessagePageQueryByGroupId

调用 OnsDLQMessagePageQueryByGroupId 分页查询指定时间段内指定 Group ID 内存在的所有死信消息。

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：

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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsDLQMessagePageQueryByGroupId	系统规定参数。取值：OnsDLQMessagePageQueryByGroupId。
BeginTime	Long	是	1570723200000	查询范围的起始时间戳。
CurrentPage	Integer	是	2	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。
EndTime	Long	是	1570809600000	查询范围的终止时间戳。
GroupId	String	是	GID_test_group_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_111111_111111_DOxxxxxx	需查询的消息所对应的实例 ID。
TaskId	String	否	0BC1310300002A9F000021E4D7A48346	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。
PageSize	Integer	否	5	分页查询，每页最多显示消息数量，默认显示 20 条，最小 5 条，最多 50 条。

② 说明 更多参数说明请参见公共参数和调用方式。

返回数据

名称	类型	示例值	描述
MsgFoundDo	Struct		查询结果。
CurrentPage	Long	1	当前位于第几页。
MaxPageCount	Long	400	查询结果最大页数。

名称	类型	示例值	描述
MsgFoundList	Array		当前页的消息集合，该类型与 OnsDLQMessageGetById 的返回结果一致。
OnsRestMessageDo			
BodyCRC	Integer	914112295	消息体 CRC 校验值。
BornHost	String	42.120.***.***:59270	生成该消息的生产者实例。
BornTimestamp	Long	1570760999721	消息生产的时间戳。
InstanceId	String	MQ_INST_111111111111_DOxxxxxx	实例 ID。
MsgId	String	1E0578FE110F18B4AAC235C05F2*****	消息 ID，即 Message ID。
PropertyList	Array		消息属性列表。
MessageProperty			
Name	String	TAGS	消息属性。取值说明如下： • TRACE_ON：是否有消息轨迹 • KEYS：消息的 Key 属性 • TAGS：消息的 Tag 属性 • INSTANCE_ID：消息所在的实例的 ID 更多概念信息请参见名词解释。
Value	String	TagA	消息属性的取值。
ReconsumeTimes	Integer	1	消息重试消费的次数。
StoreHost	String	11.193.***.***:10911	存储该消息的服务器实例。
StoreSize	Integer	406	消息大小。
StoreTimestamp	Long	1570760999811	消息被服务端存储的时间戳。

名称	类型	示例值	描述
Topic	String	test-mq_topic	消息的 Topic。
TaskId	String	0BC1310300002A9F000021E4D7A48346	第一次新建查询时，返回的查询任务 ID，用于后续取消息。
RequestId	String	B00CD3C8-D81E-4A41-85E2-38F19252F1FF	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsDLQMessagePageQueryByGroupId
&BeginTime=1570723200000
&CurrentPage=2
&EndTime=1570809600000
&GroupId=GID_test_group_id
&InstanceId=MQ_INST_111111111111_DOxxxxxx
&TaskId=0BC1310300002A9F000021E4D7A48346
&PageSize=5
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsDLQMessagePageQueryByGroupIdResponse>
  <MsgFoundDo>
    <MaxPageCount>400</MaxPageCount>
    <MsgFoundList>
      <OnsRestMessageDo>
        <MsgId>1E0578FE110F18B4AAC235C05F2*****</MsgId>
        <Instanceld>MQ_INST_111111111111_DOxxxxxx</Instanceld>
        <ReconsumeTimes>1</ReconsumeTimes>
        <BornHost>42.120.***.***:59270</BornHost>
        <StoreTimestamp>1570760999811</StoreTimestamp>
        <StoreHost>11.193.***.***:10911</StoreHost>
        <BodyCRC>914112295</BodyCRC>
        <Topic>test-mq_topic</Topic>
        <BornTimestamp>1570760999721</BornTimestamp>
        <StoreSize>406</StoreSize>
        <PropertyList>
          <MessageProperty>
            <Name>TAGS</Name>
            <Value>TagA</Value>
          </MessageProperty>
        </PropertyList>
      </OnsRestMessageDo>
    </MsgFoundList>
    <TaskId>0BC1310300002A9F000021E4D7A48346</TaskId>
    <CurrentPage>1</CurrentPage>
  </MsgFoundDo>
  <RequestId>5DC2A47E-2B31-4722-96C8-FA59C9*****</RequestId>
</OnsDLQMessagePageQueryByGroupIdResponse>
```

JSON 格式

```
{
  "OnsDLQMessagePageQueryByGroupIdResponse": {
    "MsgFoundDo": {
      "MaxPageCount": 400,
      "MsgFoundList": {
        "OnsRestMessageDo": {
          "MsgId": "1E0578FE110F18B4AAC235C05F2*****",
          "InstancelId": "MQ_INST_111111111111_DOxxxxxx",
          "ReconsumeTimes": 1,
          "BornHost": "42.120.***.***:59270",
          "StoreTimestamp": 1570760999811,
          "StoreHost": "11.193.***.***:10911",
          "BodyCRC": 914112295,
          "Topic": "test-mq_topic",
          "BornTimestamp": 1570760999721,
          "StoreSize": 406,
          "PropertyList": {
            "MessageProperty": [
              {
                "Name": "TAGS",
                "Value": "TagA"
              }
            ]
          }
        },
        "TaskId": "0BC1310300002A9F000021E4D7A48346",
        "CurrentPage": 1
      },
      "RequestId": "5DC2A47E-2B31-4722-96C8-FA59C9*****"
    }
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

20.3. OnsDLQMessageResendById

使用 OnsDLQMessageResendById 重发指定 Message ID 的死信消息，使该消息能够被 Consumer 再次消费。

本接口让死信消息重新投递，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。

?

说明

- 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。
- 当前消息队列 RocketMQ 版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。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	OnsDLQMessageResendById	系统规定参数。取值： OnsDLQMessageResendById 。
GroupId	String	是	GID_test_group_id	需查询的消费端 Group ID。
InstanceId	String	是	MQ_INST_188077086902****_BXSUW61e	需查询消息所对应的实例 ID。
MsgId	String	是	0BC16699343051CD9F1D798E7734***	需查询的消息的 ID，即 Message ID。

返回数据

名称	类型	示例值	描述
Data	List	0BC16699343051CD9F1D798E7734****	消息 ID，即 Message ID。
RequestId	String	D94CC769-4DC3-4690-A868-9D0631B1****	为公共参数，每个请求的 ID 都是唯一的。

示例

请求示例

```
http(s)://ons.cn-hangzhou.aliyuncs.com/?Action=OnsDLQMessageResendById
&GroupId=GID_test_group_id
&InstanceId=MQ_INST_188077086902****_BXSUW61e
&MsgId=0BC16699343051CD9F1D798E7734****
&<公共请求参数>
```

正常返回示例

XML 格式

```
<OnsDLQMessageResendByIdResponse>
  <Data>0BC16699343051CD9F1D798E7734****</Data>
  <RequestId>D94CC769-4DC3-4690-A868-9D0631B1****</RequestId>
</OnsDLQMessageResendByIdResponse>
```

JSON 格式

```
{
  "Data": [
    "0BC16699343051CD9F1D798E7734****"
  ],
  "RequestId": "D94CC769-4DC3-4690-A868-9D0631B1****"
}
```

错误码

访问[错误中心](#)查看更多错误码。