

阿里云

消息队列RabbitMQ版
API参考

文档版本：20220705

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.API概览	06
2.调用方式	08
3.服务接入点	09
4.获取SDK	11
5.签名机制	12
6.公共参数	15
7.获取AccessKey	17
8.实例	19
8.1. CreateInstance	19
8.2. ListInstances	24
8.3. GetMetadataAmount	28
8.4. RenewInstance	30
8.5. ModifyInstance	32
8.6. UpdateInstanceName	35
9.Vhost	38
9.1. CreateVirtualHost	38
9.2. ListVirtualHosts	39
9.3. DeleteVirtualHost	41
10.Queue	43
10.1. CreateQueue	43
10.2. ListQueues	45
10.3. ListQueueConsumers	48
10.4. DeleteQueue	50
11.Exchange	53
11.1. CreateExchange	53
11.2. ListExchanges	55

11.3. DeleteExchange	58
12.Binding	60
12.1. CreateBinding	60
12.2. ListBindings	62
12.3. ListDownStreamBindings	65
12.4. ListExchangeUpStreamBindings	69
12.5. ListQueueUpStreamBindings	72
12.6. DeleteBinding	75
13.云监控资源报表接口	78
13.1. 云监控资源报表接口说明	78
13.2. DescribeMetricMetaList	80
13.3. DescribeMetricLast	88
13.4. DescribeMetricList	94
13.5. DescribeMetricData	101
13.6. DescribeMetricTop	107

1.API概览

消息队列RabbitMQ版提供以下相关API接口。

实例

API	描述
CreateInstance	调用CreateInstance创建消息队列RabbitMQ版实例资源，调用接口会创建一个新购买实例资源的订单，并自动支付。
ListInstances	调用ListInstances查询消息队列RabbitMQ版实例。
GetMetadataAmount	调用GetMetadataAmount获取消息队列RabbitMQ版实例的元数据。
RenewInstance	调用RenewInstance对消息队列RabbitMQ版实例进行续费。
ModifyInstance	调用ModifyInstance对消息队列RabbitMQ版实例资源配置进行变更。调用接口会创建一个变配实例资源的订单，并自动支付。

Vhost

API	描述
CreateVirtualHost	调用CreateVirtualHost创建Vhost。
DeleteVirtualHost	调用DeleteVirtualHost删除Vhost。
ListVirtualHosts	调用ListVirtualHosts查询实例的Vhost。

Queue

API	描述
CreateQueue	调用CreateQueue创建Queue。
DeleteQueue	调用DeleteQueue删除Queue。
ListQueueConsumers	调用ListQueueConsumers查询Queue的消费者。
ListQueues	调用ListQueues查询Queue。

Exchange

API	描述
CreateExchange	调用CreateExchange创建Exchange。
DeleteExchange	调用DeleteExchange删除Exchange。
ListExchanges	调用ListExchanges查询Exchange。

Binding

API	描述
CreateBinding	调用CreateBinding创建绑定关系。
DeleteBinding	调用DeleteBinding删除绑定关系。
List Bindings	调用ListBindings查询某Vhost的绑定关系。
List DownStreamBindings	调用ListDownStreamBindings查询Exchange绑定情况。
List ExchangeUpStreamBindings	调用ListExchangeUpStreamBindings查询Exchange被绑定情况。
List QueueUpStreamBindings	调用ListQueueUpStreamBindings查询Queue被绑定情况。

2.调用方式

消息队列RabbitMQ版支持HTTP调用、SDK调用和OpenAPI Explorer调用。

HTTP调用

消息队列RabbitMQ版HTTP调用是向消息队列RabbitMQ版API的服务端地址发送HTTP请求，并按照接口说明在请求中加入相应请求参数，调用后系统会返回处理结果。请求及返回结果都使用UTF-8字符集进行编码。消息队列RabbitMQ版的API是RPC风格，您可以通过发送HTTP GET 请求调用消息队列RabbitMQ版的API。

其请求结构如下：

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

其中：

- Endpoint：消息队列RabbitMQ版API的服务接入地址。详情请参见[服务接入点](#)。
- Action：要执行的操作，如调用List Instances查询已创建的消息队列RabbitMQ版实例。
- Version：要使用的API版本，消息队列RabbitMQ版的API版本是2019-12-12。
- 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

消息队列RabbitMQ版提供多语言SDK。阿里云SDK免去您手动签名的过程。详情请参见[获取SDK](#)。

OpenAPI Explorer调用

OpenAPI Explorer是可视化的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](#)。

3. 服务接入点

本文列出消息队列Rabbit MQ版的API接入地域ID和接入点。

地域	地域ID (RegionId)	接入点 (Endpoint)
华东1 (杭州)	cn-hangzhou	amqp-open.cn-hangzhou.aliyuncs.com
华东2 (上海)	cn-shanghai	amqp-open.cn-shanghai.aliyuncs.com
华北1 (青岛)	cn-qingdao	amqp-open.cn-qingdao.aliyuncs.com
华北2 (北京)	cn-beijing	amqp-open.cn-beijing.aliyuncs.com
华北3 (张家口)	cn-zhangjiakou	amqp-open.cn-zhangjiakou.aliyuncs.com
华北5 (呼和浩特)	cn-huhehaote	amqp-open.cn-huhehaote.aliyuncs.com
华南1 (深圳)	cn-shenzhen	amqp-open.cn-shenzhen.aliyuncs.com
华南3 (广州)	cn-guangzhou	amqp-open.cn-guangzhou.aliyuncs.com
西南1 (成都)	cn-chengdu	amqp-open.cn-chengdu.aliyuncs.com
中国香港	cn-hongkong	amqp-open.cn-hongkong.aliyuncs.com
印度尼西亚 (雅加达)	ap-southeast-5	amqp-open.ap-southeast-5.aliyuncs.com
日本 (东京)	ap-northeast-1	amqp-open.ap-northeast-1.aliyuncs.com
新加坡 (新加坡)	ap-southeast-1	amqp-open.ap-southeast-1.aliyuncs.com
澳大利亚 (悉尼)	ap-southeast-2	amqp-open.ap-southeast-2.aliyuncs.com
马来西亚 (吉隆坡)	ap-southeast-3	amqp-open.ap-southeast-3.aliyuncs.com
德国 (法兰克福)	eu-central-1	amqp-open.eu-central-1.aliyuncs.com

地域	地域ID (RegionId)	接入点 (Endpoint)
美国 (硅谷)	us-west-1	amqp-open.us-west-1.aliyuncs.com
美国 (弗吉尼亚)	us-east-1	amqp-open.us-east-1.aliyuncs.com
华东2金融云	cn-shanghai-finance-1	amqp-open.cn-shanghai-finance-1.aliyuncs.com
华北2金融云	cn-beijing-finance-1	amqp-open.cn-beijing-finance-1.aliyuncs.com

4. 获取SDK

本文介绍如何获取和使用消息队列Rabbit MQ版多语言SDK。

SDK下载

消息队列Rabbit MQ版的多语言SDK的下载地址及说明文档如下。

语言	下载地址	说明文档
Java	Alibaba Cloud AMQP SDK for Java	开始使用
Go	Alibaba Cloud AMQP SDK for Go	开始使用
Python	Alibaba Cloud AMQP SDK for Python	开始使用
PHP	Alibaba Cloud AMQP for PHP	开始使用

5. 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=xxxx%xxxx%3D&SignatureNonce=e7b1f31150be45594905ce6d28561286
```

其中：

- 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。用户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（Universally Unique Identifier, UUID）。
- 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

1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）。
 - i. 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 <code>%XY</code> 的格式，其中 <code>XY</code> 是字符对应ASCII码的16进制表示。例如英文的双引号（ <code>" "</code> ）对应的编码为 <code>%22</code> 。
扩展的UTF-8字符	编码成 <code>%XY%ZA...</code> 的格式。
英文空格	编码成 <code>%20</code>，而不是加号（<code>+</code>）。 该编码方式和一般采用的 <code>application/x-www-form-urlencoded</code> MIME格式编码算法（例如Java标准库中的 <code>java.net.URLEncoder</code> 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（<code>+</code>）替换成 <code>%20</code>，星号（<code>*</code>）替换成 <code>%2A</code>，<code>%7E</code> 替换回波浪号（<code>~</code>），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 <code>percentEncode</code> 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成待签名的字符串。

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

其中：

- `HTTPMethod` 是提交请求用的HTTP方法，例如GET。
- `percentEncode(“/”)` 是按照步骤1.1中描述的URL编码规则对字符“/”进行编码得到的值，即`%2F`。

- percentEncode(CanonicalizedQueryString) 是对步骤1中构造的规范化请求字符串按步骤1.2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ListInstances API为例，假设使用的 AccessKey Id 为 testid ， AccessKey Secret 为 testsecret 。签名前的请求URL如下：

```
http://amqp-open.aliyuncs.com/?Timestamp=2020-02-11T06%3A00%3A10Z&Format=JSON&AccessKeyId=testid&Action=ListInstances&SignatureMethod=HMAC-SHA1&SignatureNonce=1e428a4d3e45bce88b2ed41cc34497eb&Version=2019-12-12&SignatureVersion=1.0
```

使用 testsecret& ，计算得到的签名值是：

```
xtC1HBet6MjHTQVZLo%2Fbm4dlJ0s%3D
```

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://amqp-open.aliyuncs.com/?SignatureVersion=1.0&Action=ListInstances&Format=JSON&SignatureNonce=1e428a4d3e45bce88b2ed41cc34497eb&Version=2019-12-12&AccessKeyId=testid&Signature=xtC1HBet6MjHTQVZLo%2Fbm4dlJ0s%3D&SignatureMethod=HMAC-SHA1&Timestamp=2020-02-11T06%3A00%3A10Z
```

6. 公共参数

介绍每个接口都需要使用的请求参数和返回参数。

公共请求参数

名称	类型	是否必选	描述
Format	String	否	返回消息的格式。取值： <i>JSON（默认值）/XML</i>
Version	String	是	API版本号，使用YYYY-MM-DD日期格式。取值： <i>2019-12-12</i>
AccessKeyId	String	是	访问服务使用的密钥ID。
Signature	String	是	签名结果串。
SignatureMethod	String	是	签名方式，取值： <i>HMAC-SHA1</i>
Timestamp	String	是	请求的时间戳，为日期格式。使用UTC时间按照ISO8601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： <i>1.0</i>
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，建议使用通用唯一识别码（Universally Unique Identifier, UUID）。
ResourceOwnerAccount	String	否	本次API请求访问到的资源拥有者账户，即登录用户名。

示例

```
https://amqp-open.aliyuncs.com/?Action=ListInstances
&Format=JSON
&Version=2019-12-12
&Signature=xxxx%xxxx%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=e7b1f31150be45594905ce6d28561286
&SignatureVersion=1.0
&AccessKeyId=key-test
&TimeStamp=2020-02-10T08%3A43%3A05Z
...
```

公共返回参数

API返回结果采用统一格式，调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为JSON格式。每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

- 返回 `2xx` HTTP状态码表示调用成功。
- 返回 `4xx` 或 `5xx` HTTP状态码表示调用失败。

公共返回参数示例如下：

- XML格式

```
<?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>D6660A81-1A96-43F3-AC00-4EE473C70AEC</RequestId>
    <!--返回结果数据-->
  </接口名称+Response>
```

- JSON格式

```
{
  "RequestId": "D6660A81-1A96-43F3-AC00-4EE473C70AEC",
  /*返回结果数据*/
}
```


7. 获取AccessKey

您可以为阿里云账号（主账号）和RAM用户创建一个访问密钥（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。

- AccessKey ID：用于标识用户。
- AccessKey Secret：用于验证用户的密钥。AccessKey Secret必须保密。

 **警告** 阿里云账号AccessKey泄露会威胁您所有资源的安全。建议使用RAM用户AccessKey进行操作，可以有效降低AccessKey泄露的风险。

操作步骤

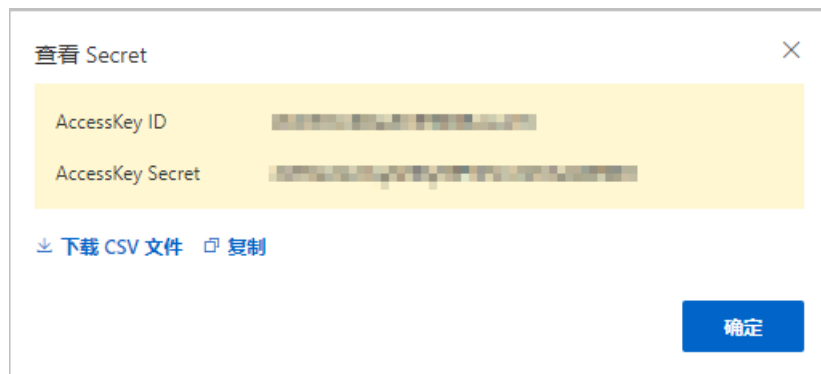
1. 使用阿里云账号登录[阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击**AccessKey管理**。
3. 在安全提示对话框，选择使用阿里云账号AccessKey或RAM用户AccessKey。



4. 获取账号AccessKey：
 - 获取阿里云账号AccessKey
 - a. 单击**继续使用AccessKey**。
 - b. 在AccessKey管理页面，单击**创建AccessKey**。
 - c. 在**手机验证**对话框，获取验证码，完成手机验证，单击**确定**。

- d. 在创建AccessKey对话框，查看AccessKey ID和AccessKey Secret。

您可以单击下载CSV文件，下载AccessKey信息。或者单击复制，复制AccessKey信息。



- o 获取RAM用户AccessKey

- 单击开始使用子用户AccessKey。
- (可选) 如果未创建RAM用户，在系统跳转的RAM访问控制台的创建用户页面，创建RAM用户。
如果是获取已有RAM用户的AccessKey，则跳过此步骤。
- 在左侧导航栏，选择人员管理 > 用户，搜索需要获取AccessKey的用户。
- 单击用户登录名称。
- 在认证管理页签下的用户AccessKey区域，单击创建AccessKey。
- 在手机验证页面，获取验证码，完成手机验证，单击确定。
- 在创建AccessKey页面，查看AccessKey ID和AccessKey Secret。

您可以单击下载CSV文件，下载AccessKey信息。或者单击复制，复制AccessKey信息。



8. 实例

8.1. CreateInstance

调用CreateInstance创建一个实例资源，调用接口会创建一个新购实例资源的订单，并自动支付。

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	CreateInstance	系统规定参数。取值：CreateInstance。
Parameter.N.Code	String	是	InstanceType	第N个模块属性Code配置，如有多个模块属性参数，按照N=0,1,2,3...N的形式拼接多个参数。 该参数需与Parameter.N.Value一起使用，消息队列RabbitMQ版创建实例还需要配置更多属性，请参见 配置Parameter.N.Code参数 。
Parameter.N.Value	String	是	professional	第N个模块属性Code对应的Value，如有多个模块属性参数，按照N=0,1,2,3...N的形式拼接多个参数。
ProductCode	String	是	ons	产品代码，固定为：ons。
ProductType	String	是	ons_onsproxy_pre	产品类型，固定为：ons_onsproxy_pre。
SubscriptionType	String	是	Subscription	付费类型，取值如下： - Subscription: 预付费 - PayAsYouGo: 后付费 消息队列RabbitMQ版实例不再支持按量付费模式，即该参数只能为Subscription。
RenewalStatus	String	否	AutoRenewal	续费方式，取值如下： - AutoRenewal: 自动续费 - ManualRenewal: 手动续费

名称	类型	是否必选	示例值	描述
RenewPeriod	Integer	否	1	自动续费周期，单位：月。 ❓ 说明 当RenewalStatus取值为AutoRenewal时，该参数必须设置。
Period	Integer	否	1	预付费周期，单位：月。按年付费产品请输入12的整数倍。 ❓ 说明 当创建预付费实例时，必须设置。
ClientToken	String	否	JASIOFKVNVIXXX XX	客户端幂等参数，服务端会查询是否有相同ClientToken的请求，如果有，直接返回上次调用结果。

配置Parameter.N.Code参数

参数	取值	是否必选	描述
Parameter.0.Code	InstanceType	否	配置实例规格类型。Parameter.0.Value取值如下： - professional: 专业版 - enterprise: 企业版 - VIP: 铂金版
Parameter.1.Code	Region	是	配置实例所属地域。Parameter.1.Value取值为所支持的地域，例如cn-shenzhen。
Parameter.2.Code	SupportEip	是	配置实例是否支持公网。Parameter.2.Value取值如下： - eip_true: 支持 - eip_false: 不支持
Parameter.3.Code	MaxEipTps	是	配置公网TPS流量峰值。如何配置Parameter.3.Value，请根据 消息队列RabbitMQ版控制台 购买页所给出的取值设置。

参数	取值	是否必选	描述
Parameter.4.Code	MaxTps	是	配置私网TPS流量峰值。如何配置Parameter.4.Value, 请根据 消息队列RabbitMQ版控制台 购买页所给出的取值设置。
Parameter.5.Code	QueueCapacity	是	配置Queue的数量上限。Parameter.5.Value的取值范围为[50,6000], 单位: 个。
Parameter.6.Code	OrderNum	是	配置创建实例的数量。Parameter.6.Value取值范围为[1,9999], 单位: 个。
Parameter.7.Code	StorageSize	是	配置消息存储空间。Parameter.7.Value取值如下: - 专业版实例和企业版实例: 固定为200, 单位: GB。 - 铂金版实例: m×100, 其中m的取值范围为[7,28], 单位: GB。
Parameter.8.Code	SupportTracing	是	配置是否开通消息轨迹功能。Parameter.8.Value取值如下: - tracing_true: 表示开通消息轨迹功能。 - tracing_false: 表示不开通消息轨迹功能。  注意 - 铂金版实例免费提供15天消息轨迹功能, 请参见下文消息轨迹参数取值约束。

参数	取值	是否必选	描述
Parameter.9.Code	TracingStorageTime	是	配置消息轨迹的保存时长。Parameter.9.Value取值如下： • 0 • 3 • 7 • 15 单位：天。  注意 Parameter.9.Value参数必须和Parameter.8.Value参数配合使用。具体限制，请参见下文消息轨迹参数取值约束。

消息轨迹参数取值约束

InstanceType参数取值	SupportTracing参数取值	TracingStorageTime参数取值	组合效果
• professional • enterprise	tracing_false	0	创建专业版或企业版实例且不开启消息轨迹功能时，消息轨迹的保存时长只能设置为0天。
• professional • enterprise	tracing_true	• 3 • 7 • 15	开启消息轨迹功能，且实例的规格类型为专业版或企业版时，消息轨迹的保存时长可以设置为3天、7天或15天。
VIP	tracing_true	15	实例规格类型为铂金版时，消息队列RabbitMQ版免费提供15天的消息轨迹保存功能。建议开启消息轨迹功能，并将消息轨迹保存时长设置为15天。

返回数据

名称	类型	示例值	描述
Code	String	Success	状态码。

名称	类型	示例值	描述
Data	Struct		返回信息。
InstanceId	String	amqp-cn-zvp2ajsj****	订单对应的实例ID。
OrderId	String	21014976059****	创建成功的订单ID。
Message	String	Successful!	提示信息。
RequestId	String	B62C712B-1789-5114-B0BB-80CB2AFD****	请求ID。
Success	Boolean	true	是否成功。

示例

请求示例

```
http(s)://business.aliyuncs.com/?Action=CreateInstance
&Parameter.0.Code=InstanceType
&Parameter.0.Value=professional
&Parameter.1.Code=Region
&Parameter.1.Value=cn-shenzhen
&Parameter.2.Code=SupportEip
&Parameter.2.Value=eip_true
&Parameter.3.Code=MaxEipTps
&Parameter.3.Value=640
&Parameter.4.Code=MaxTps
&Parameter.4.Value=1000
&Parameter.5.Code=QueueCapacity
&Parameter.5.Value=200
&Parameter.6.Code=StorageSize
&Parameter.6.Value=200
&Parameter.7.Code=OrderNum
&Parameter.7.Value=1
&Parameter.8.Code=SupportTracing
&Parameter.8.Value=tracing_true
&Parameter.9.Code=TracingStorageTime
&Parameter.9.Value=3
&ProductCode=ons
&ProductType=ons_onsproxy_pre
&SubscriptionType=Subscription
&Period=1
&RenewPeriod=1
&RenewalStatus=AutoRenewal
&<公共请求参数>
```

正常返回示例

XML

格式

```
<CreateInstanceResponse>
  <RequestId>B62C712B-1789-5114-B0BB-80CB2AFD****</RequestId>
  <Message>Successful!</Message>
  <Data>
    <InstanceId>amqp-cn-zvp2ajsj****</InstanceId>
    <OrderId>21014976059****</OrderId>
  </Data>
  <Code>Success</Code>
  <Success>true</Success>
</CreateInstanceResponse>
```

JSON

格式

```
{
  "RequestId": "B62C712B-1789-5114-B0BB-80CB2AFD****",
  "Message": "Successful!",
  "Data": {
    "InstanceId": "amqp-cn-zvp2ajsj****",
    "OrderId": "21014976059****"
  },
  "Code": "Success",
  "Success": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2. ListInstances

调用ListInstances查询实例。

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
Action	String	是	ListInstances	要执行的操作。取值： ListInstances。
MaxResults	Integer	是	1	最大结果集。建议取值在1~100之间。

名称	类型	是否必选	示例值	描述
RegionId	String	是	cn-hangzhou	实例的地域ID。
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Data	Struct		数据。
Instances	Array of InstanceVO		实例。
AutoRenewInstance	Boolean	false	是否自动更新实例。
ClassicEndpoint	String	amqp-cn-st21x7kv****.not-support	经典接入点，该参数不再使用。
ExpireTime	Long	1651507200000	过期时间戳，单位：毫秒。
InstanceId	String	amqp-cn-st21x7kv****	实例ID。
InstanceName	String	amqp-cn-st21x7kv****	实例名称。
InstanceType	String	professional	实例类型。 - professional：专业版 - enterprise：企业版 - VIP：铂金版
MaxEipTps	Integer	24832	实例的公网TPS流量峰值，单位：条/秒。
MaxQueue	Integer	50	实例的Queue的数量上限，单位：个。

名称	类型	示例值	描述
MaxTps	Integer	5000	实例的私网TPS流量峰值，单位：条/秒。
MaxVhost	Integer	50	实例的Vhost的数量上限，单位：个。
OrderCreateTime	Long	1572441939000	订单创建时间戳，单位：毫秒。
OrderType	String	PRE_PAID	订单类型。
PrivateEndpoint	String	amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com	VPC接入点。
PublicEndpoint	String	amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com	公网接入点。
Status	String	SERVING	实例状态。
StorageSize	Integer	200	磁盘容量，单位：GB。 ② 说明 专业版实例和企业版实例无此参数，返回-1。
SupportEIP	Boolean	true	是否支持EIP。
MaxResults	Integer	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
RequestId	String	CCBB1225-C392-480E-8C7F-D09AB2CD2***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListInstances
&MaxResults=1
&RegionId=cn-hangzhou
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListInstancesResponse>
  <RequestId>CCBB1225-C392-480E-8C7F-D09AB2CD2***</RequestId>
  <Data>
    <Instances>
      <Status>SERVING</Status>
      <MaxTps>5000</MaxTps>
      <MaxEipTps>24832</MaxEipTps>
      <ClassicEndpoint>amqp-cn-st21x7kv****.not-support</ClassicEndpoint>
      <InstanceId>amqp-cn-st21x7kv****</InstanceId>
      <MaxQueue>50</MaxQueue>
      <SupportEIP>true</SupportEIP>
      <StorageSize>200</StorageSize>
      <InstanceName>amqp-cn-st21x7kv****</InstanceName>
      <OrderType>PRE_PAID</OrderType>
      <OrderCreateTime>1572441939000</OrderCreateTime>
      <MaxVhost>50</MaxVhost>
      <PublicEndpoint>amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com</PublicEndpoint>
      <InstanceType>professional</InstanceType>
      <ExpireTime>1651507200000</ExpireTime>
      <AutoRenewInstance>false</AutoRenewInstance>
      <PrivateEndpoint>amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com</PrivateEndpoint>
    </Instances>
    <NextToken></NextToken>
    <MaxResults>1</MaxResults>
  </Data>
</ListInstancesResponse>
```

JSON 格式

```
{
  "RequestId": "CCBB1225-C392-480E-8C7F-D09AB2CD2***",
  "Data": {
    "Instances": {
      "Status": "SERVING",
      "MaxTps": 5000,
      "MaxEipTps": 24832,
      "ClassicEndpoint": "amqp-cn-st21x7kv****.not-support",
      "InstanceId": "amqp-cn-st21x7kv****",
      "MaxQueue": 50,
      "SupportEIP": true,
      "StorageSize": 200,
      "InstanceName": "amqp-cn-st21x7kv****",
      "OrderType": "PRE_PAID",
      "OrderCreateTime": 1572441939000,
      "MaxVhost": 50,
      "PublicEndpoint": "amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com",
      "InstanceType": "professional",
      "ExpireTime": 1651507200000,
      "AutoRenewInstance": false,
      "PrivateEndpoint": "amqp-cn-st21x7kv****.mq-amqp.cn-hangzhou-a.aliyuncs.com"
    },
    "NextToken": "",
    "MaxResults": 1
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.3. GetMetadataAmount

调用GetMetadataAmount获取实例的元数据。

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	GetMetadataAmount	要执行的操作。取值： GetMetadataAmount 。
RegionId	String	否	cn-hangzhou	地域ID。

名称	类型	是否必选	示例值	描述
InstanceId	String	否	amqp-cn-v0h1kb9n***	实例ID。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
CurrentExchanges	Integer	7	已创建的Exchange的数量。
CurrentQueues	Integer	1	已创建的Queue的数量。
CurrentVirtualHosts	Integer	1	已创建的Vhost的数量。
MaxExchanges	Integer	20	可创建的Exchange的最大数量。
MaxQueues	Integer	20	可创建的Queue的最大数量。
MaxVirtualHosts	Integer	10	可创建的Vhost的最大数量。
RequestId	String	B75ACF23-2BEB-44AC-A0B6-AE14EDCA***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=GetMetadataAmount
&InstanceId=amqp-cn-v0h1kb9n***
&RegionId=cn-hangzhou
&<公共请求参数>
```

正常返回示例

XML 格式

```
<GetMetadataAmountResponse>
  <RequestId>B75ACF23-2BEB-44AC-A0B6-AE14EDCA***</RequestId>
  <Data>
    <MaxExchanges>20</MaxExchanges>
    <MaxVirtualHosts>10</MaxVirtualHosts>
    <CurrentExchanges>7</CurrentExchanges>
    <MaxQueues>20</MaxQueues>
    <CurrentVirtualHosts>1</CurrentVirtualHosts>
    <CurrentQueues>1</CurrentQueues>
  </Data>
</GetMetadataAmountResponse>
```

JSON 格式

```
{
  "RequestId": "B75ACF23-2BEB-44AC-A0B6-AE14EDCA***",
  "Data": {
    "MaxExchanges": 20,
    "MaxVirtualHosts": 10,
    "CurrentExchanges": 7,
    "MaxQueues": 20,
    "CurrentVirtualHosts": 1,
    "CurrentQueues": 1
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.4. RenewInstance

调用RenewInstance对相关实例进行续费。

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	RenewInstance	系统规定参数。取值：RenewInstance。
InstanceId	String	是	amqp-cn-zvp2ajsj****	实例ID。
ProductCode	String	是	ons	产品代码，固定为：ons。

名称	类型	是否必选	示例值	描述
RenewPeriod	Integer	是	6	预付费续费时长，单位：月。取值如下： 1~3 6 12 24 36
ClientToken	String	否	ASKJHKLASJHAFSLKJH	用于保证请求的幂等性。
ProductType	String	否	ons_onsproxy_pre	产品类型，固定为：ons_onsproxy_pre。

返回数据

名称	类型	示例值	描述
Code	String	Success	状态码。
Data	Struct		返回数据。
OrderId	String	21016588106****	订单ID。
Message	String	Successful!	提示信息。
RequestId	String	60CE9D3C5A-4346-52B5-AC15-83CFEF73****	请求ID。
Success	Boolean	true	是否成功。

示例

请求示例

```
http(s)://business.aliyuncs.com/?Action=RenewInstance
&ProductCode=ons
&InstanceId=amqp-cn-zvp2ajsj****
&RenewPeriod=6
&ProductType=ons_onsproxy_pre
&<公共请求参数>
```

正常返回示例

XML

格式

```
<RenewInstanceResponse>
  <Message>Successful!</Message>
  <RequestId>CE9D3C5A-4346-52B5-AC15-83CFEF73****</RequestId>
  <Data>
    <OrderId>21016588106****</OrderId>
  </Data>
  <Code>Success</Code>
  <Success>true</Success>
</RenewInstanceResponse>
```

JSON

格式

```
{
  "Message": "Successful!",
  "RequestId": "CE9D3C5A-4346-52B5-AC15-83CFEF73****",
  "Data": {
    "OrderId": "21016588106****"
  },
  "Code": "Success",
  "Success": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.5. ModifyInstance

调用ModifyInstance对实例资源配置进行变更。调用接口会创建一个变配实例资源的订单，并自动支付。

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	ModifyInstance	系统规定参数。取值：ModifyInstance。
ModifyType	String	是	Upgrade	变配类型，取值 - Upgrade：升级。 - Downgrade：降级。

名称	类型	是否必选	示例值	描述
Parameter.N.Code	String	是	test	第N个模块属性code配置，如有多个模块属性参数，按照N=0,1,2,3...N的形式拼接多个参数。  说明 只设置需要变更配置的模块属性信息，如A,B两个属性中，只有A需要变更配置，则只传A属性相关即可。 该参数需与Parameter.N.Value一起使用，消息队列RabbitMQ版实例支持的变更配置包括： - 变更公网TPS流量峰值：您需设置Parameter.N.Code为 <i>MaxEipTps</i>，对应Parameter.N.Value参数，请根据消息队列RabbitMQ版控制台购买页所给出的取值设置。 - 变更私网TPS流量峰值：您需设置Parameter.N.Code为 <i>MaxTps</i>，对应Parameter.N.Value参数，请根据消息队列RabbitMQ版控制台购买页所给出的取值设置。 - 变更Queue的数量上限：您需设置Parameter.N.Code为 <i>QueueCapacity</i>，对应Parameter.N.Value的取值范围为[50,6000]，单位：个。 - 变更消息存储空间：铂金版实例支持变更消息存储空间。您需设置Parameter.N.Code为 <i>StorageSize</i>，对应Parameter.N.Value的取值为m×100，其中m的取值范围为[7,28]，单位：GB。
Parameter.N.Value	String	是	test	第N个模块属性Code对应的Value，按照N=0,1,2,3...N的形式拼接多个参数。
ProductCode	String	是	ons	产品代码，固定为：ons。
SubscriptionType	String	是	Subscription	付费类型，取值： - Subscription：预付费。 - PayAsYouGo：后付费。 消息队列RabbitMQ版实例不再支持按量付费模式，即该参数只能为<i>Subscription</i>。
ProductType	String	是	ons_onsproxy_pre	产品类型，固定为：ons_onsproxy_pre。

名称	类型	是否必选	示例值	描述
InstanceId	String	是	amqp-cn-zvp2ajsj****	变更配置的实例ID。
ClientToken	String	否	JAKSJFHFAKJSF	用于保证请求的幂等性，防止请求重复提交。

返回数据

名称	类型	示例值	描述
Code	String	Success	状态码。
Data	Struct		返回数据。
HostId	String	cn	主机ID。
OrderId	String	21017220559****	创建成功的订单ID。
Message	String	Successful!	提示信息。
RequestId	String	DFCE1EB7-6AE6-5AC9-BC3C-42B0EC5B****	请求ID。
Success	Boolean	true	是否成功。

示例

请求示例

```
http(s)://business.aliyuncs.com/?Action=ModifyInstance
&ModifyType=Upgrade
&Parameter.0.Code=MaxTps
&Parameter.0.Value=60000
&Parameter.1.Code=MaxEipTps
&Parameter.1.Value=49408
&Parameter.2.Code=QueueCapacity
&Parameter.2.Value=60
&InstanceId=amqp-cn-i7m2a8bq****
&ProductCode=ons
&&ProductType=ons_onsproxy_pre
&SubscriptionType=Subscription&<公共请求参数>
```

正常返回示例

XML 格式

```
<ModifyInstanceResponse>
  <Message>Successful!</Message>
  <RequestId>DFCE1EB7-6AE6-5AC9-BC3C-42B0EC5B****</RequestId>
  <Data>
    <OrderId>21017220559****</OrderId>
    <HostId>cn</HostId>
  </Data>
  <Code>Success</Code>
  <Success>true</Success>
</ModifyInstanceResponse>
```

JSON 格式

```
{
  "Message": "Successful!",
  "RequestId": "DFCE1EB7-6AE6-5AC9-BC3C-42B0EC5B****",
  "Data": {
    "OrderId": "21017220559****",
    "HostId": "cn"
  },
  "Code": "Success",
  "Success": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.6. UpdateInstanceName

调用UpdateInstanceName更新实例的名称。

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	UpdateInstanceName	系统规定参数。取值：UpdateInstanceName。
InstanceId	String	是	amqp-cn-zvp2ajsj****	实例ID。

名称	类型	是否必选	示例值	描述
InstanceName	String	是	amqp-cn-ZVp2ajsj****	需更新的实例名称。取值无限制，建议长度不超过64个字符。
RegionId	String	是	cn-hangzhou	实例所属地域。

返回数据

名称	类型	示例值	描述
Code	Integer	200	状态码。
Data	String	true	返回信息。
RequestId	String	6DC68EC9-0E76-5435-B8C0-FF9492B4****	请求ID。
Message	String	InstanceNotExist	当执行报错时，返回此参数，提示错误信息。
Success	Boolean	true	执行成功的返回信息。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=UpdateInstanceName
&InstanceId=amqp-cn-zvp2ajsj****
&InstanceName=amqp-cn-ZVp2ajsj****
&RegionId=cn-hangzhou
&<公共请求参数>
```

正常返回示例

XML 格式

```
<UpdateInstanceNameResponse>
  <RequestId>6DC68EC9-0E76-5435-B8C0-FF9492B4****</RequestId>
  <Data>true</Data>
  <Code>200</Code>
  <Success>true</Success>
</UpdateInstanceNameResponse>
```

JSON 格式

```
{
  "RequestId": "6DC68EC9-0E76-5435-B8C0-FF9492B4****",
  "Data": true,
  "Code": 200,
  "Success": true
}
```

错误码

HttpCode	错误码	错误信息	描述
500	InternalServerError	The request processing has failed due to some unknown error.	请求因遭遇某些未知错误而失败。

访问[错误中心](#)查看更多错误码。

9.Vhost

9.1. CreateVirtualHost

调用CreateVirtualHost创建Vhost。

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	CreateVirtualHost	要执行的操作。取值： CreateVirtualHost 。
InstanceId	String	是	amqp-cn-v0h1kb9n***	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	Demo	Vhost名称。说明： - Vhost名称只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@）。 - Vhost名称长度限制1~255字符。 - Vhost创建后，Vhost名称不可修改，只能删除重建。

返回数据

名称	类型	示例值	描述
RequestId	String	628705FD-03EE-4ABE-BB21-E1672960***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=CreateVirtualHost
&InstanceId=amqp-cn-v0h1kb9n***
&RegionId=cn-hangzhou
&VirtualHost=Demo
&<公共请求参数>
```

正常返回示例

XML 格式

```
<CreateVirtualHostResponse>
  <RequestId>628705FD-03EE-4ABE-BB21-E1672960***</RequestId>
</CreateVirtualHostResponse>
```

JSON 格式

```
{
  "RequestId": "628705FD-03EE-4ABE-BB21-E1672960***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

9.2. ListVirtualHosts

调用ListVirtualHosts查询Vhost。

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	ListVirtualHosts	要执行的操作。取值： ListVirtualHosts 。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	2	最大结果集。取值： 1 ~ 100 。
RegionId	String	是	cn-hangzhou	地域ID。

名称	类型	是否必选	示例值	描述
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
MaxResults	Integer	2	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
VirtualHosts	Array of VhostVO		Vhost。
Name	String	test	Vhost名称。
RequestId	String	EF4DB019-DA4A-4CE3-B220-223BBC93F***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListVirtualHosts
&InstanceId=1880770869023***
&MaxResults=2
&RegionId=cn-hangzhou
&<公共请求参数>
```

正常返回示例

XML 格式


```
<ListVirtualHostsResponse>
  <RequestId>EF4DB019-DA4A-4CE3-B220-223BBC93F***</RequestId>
  <Data>
    <NextToken></NextToken>
    <MaxResults>2</MaxResults>
    <VirtualHosts>
      <Name>test</Name>
    </VirtualHosts>
  </Data>
</ListVirtualHostsResponse>
```

JSON 格式

```
{
  "RequestId": "EF4DB019-DA4A-4CE3-B220-223BBC93F***",
  "Data": {
    "NextToken": "",
    "MaxResults": "2",
    "VirtualHosts": [
      {
        "Name": "test"
      }
    ]
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

9.3. DeleteVirtualHost

调用DeleteVirtualHost删除Vhost。

注意：您在调用DeleteVirtualHost删除Vhost前，请确保已删除该Vhost下的所有Exchange和Queue。

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	DeleteVirtualHost	要执行的操作。取值： DeleteVirtualHost 。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。

名称	类型	是否必选	示例值	描述
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。

返回数据

名称	类型	示例值	描述
RequestId	String	4311050D-BD63-48F9-822B-947A75A1***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteVirtualHost
&InstanceId=amqp-cn-v0h1kb9nu***
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML

格式

```
<DeleteVirtualHostResponse>
  <RequestId>4311050D-BD63-48F9-822B-947A75A1***</RequestId>
</DeleteVirtualHostResponse>
```

JSON

格式

```
{
  "RequestId": "4311050D-BD63-48F9-822B-947A75A1***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

10.Queue

10.1. CreateQueue

调用CreateQueue创建Queue。

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	CreateQueue	要执行的操作。取值： CreateQueue 。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。
QueueName	String	是	DemoQueue	队列名称。 - Queue名称只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@），长度限制1~255字符。 - Queue创建后，名称无法修改，只能删除后重建。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@），长度限制1~255字符。
AutoDeleteState	Boolean	否	false	是否自动删除。取值： - true：是。如果订阅该Queue的最后一个消费端取消订阅后，那么该Queue将会自动删除。 - false：否。

名称	类型	是否必选	示例值	描述
ExclusiveState	Boolean	否	false	是否为排他性Exchange。取值： • true：是。只能被声明该排他性Exchange的Connection使用。该Connection断开后该排他性Exchange自动删除。 • false：否。
MessageTTL	Long	否	1000	消息在Queue中的有效期。 • 某条消息在Queue中的留存时间超过配置的消息存活时间时，则该消息过期。 • 消息存活时间的值必须为非负整型数，最大为1天。单位为毫秒。例如，某条消息的存活时间的值是1000，则代表该消息最多会在Queue中存活1秒。
AutoExpireState	Long	否	10000	Queue的自动过期时间。如果Queue在指定时间内没有被访问，则会被自动删除。  说明 该参数对应的功能需要开通后才能使用。如需使用，请提交工单开通。
MaxLength	Long	否	1000	当前版本不支持该参数。 Queue中消息的最大数量。如果超出该数量，则先到达该Queue的消息将会被删除。
DeadLetterExchange	String	否	DLExchange	死信Exchange。该类Exchange是用来接收被拒绝的消息。 如果消费端拒绝一个不重新发送的消息，那么消息队列RabbitMQ版将会把消息路由到指定的死信Exchange，该Exchange再将消息路由到绑定的Queue进行存储。
DeadLetterRoutingKey	String	否	test.dl	死信Routing Key。 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@），长度限制1~255字符。
MaximumPriority	Integer	否	10	不支持优先级功能。取值不影响调用和返回结果。

返回数据

名称	类型	示例值	描述
RequestId	String	59B52E2C-0B8E-44EC-A314-D0314A50***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=CreateQueue
&InstanceId=amqp-cn-v0h1kb9nu***
&MaxLength=1000
&QueueName=DemoQueue
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<CreateQueueResponse>
  <RequestId>59B52E2C-0B8E-44EC-A314-D0314A50***</RequestId>
</CreateQueueResponse>
```

JSON 格式

```
{
  "RequestId": "59B52E2C-0B8E-44EC-A314-D0314A50***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

10.2. ListQueues

调用ListQueues查询Queue。

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	ListQueues	要执行的操作。取值： ListQueues 。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	1	最大结果集。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a* ***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
MaxResults	Integer	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
Queues	Array of QueueVO		Queue。
Attributes	Map	test	属性。
AutoDeleteState	Boolean	false	自动删除状态。
CreateTime	Long	1580887085240	创建时间。
ExclusiveState	Boolean	false	是否排外的。

名称	类型	示例值	描述
LastConsumeTime	Long	1680887085240	上一次消费时间。
Name	String	QueueTest	Queue名称。
OwnerId	String	1880770869023***	实例ID。
VHostName	String	test	Vhost名称。
RequestId	String	CE811989-9F02-42CE-97A6-2239CB5C2***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.co/?Action=ListQueues
&InstanceId=1880770869023***
&MaxResults=1
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListQueuesResponse>
  <RequestId>CE811989-9F02-42CE-97A6-2239CB5C2***</RequestId>
  <Data>
    <NextToken>,</NextToken>
    <MaxResults>1</MaxResults>
    <Queues>
      <VHostName>test</VHostName>
      <OwnerId>1880770869023***</OwnerId>
      <CreateTime>1580887085240</CreateTime>
      <ExclusiveState>>false</ExclusiveState>
      <Attributes>test</Attributes>
      <LastConsumeTime>1680887085240</LastConsumeTime>
      <AutoDeleteState>>false</AutoDeleteState>
      <Name>QueueTest</Name>
    </Queues>
  </Data>
</ListQueuesResponse>
```

JSON 格式

```
{
  "RequestId": "CE811989-9F02-42CE-97A6-2239CB5C2***",
  "Data": {
    "NextToken": "",
    "MaxResults": 1,
    "Queues": {
      "VHostName": "test",
      "OwnerId": "1880770869023***",
      "CreateTime": 1580887085240,
      "ExclusiveState": false,
      "Attributes": "test",
      "LastConsumeTime": 1680887085240,
      "AutoDeleteState": false,
      "Name": "QueueTest"
    }
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

10.3. ListQueueConsumers

调用ListQueueConsumers查询Queue的消费者。

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	ListQueueConsumers	要执行的操作。取值： ListQueueConsumers。
InstanceId	String	是	188077086902***	实例ID。
Queue	String	是	queue-rabbit-springboot-advance5	队列名称。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。

名称	类型	是否必选	示例值	描述
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。
QueryCount	Integer	否	1	拉取数据的条数。如果不设置，默认值为1。 本参数取值无限制，建议设置在[1,100]。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
Consumers	Array of QueueConsumerVO		消费者信息。
ConsumerTag	String	sgen-1	消费者标签。
MaxResults	Integer	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
RequestId	String	4409B7D5-E4EC-4EB5-804A-385DCDFCD***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListQueueConsumers
&InstanceId=188077086902***
&Queue=queue-rabbit-springboot-advance5
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListQueueConsumersResponse>
  <RequestId>4409B7D5-E4EC-4EB5-804A-385DCDFCD***</RequestId>
  <Data>
    <NextToken>AAAAATzkBMTg4MDc3MDg2OTAyMzQyMAF0ZXN0AXF1ZXVlLXJhYmJpdC1zcHJpbmdib290LW
    FkdmFuY2U1ATc3NjMxNjIwATE1OTEzNTA5MDkzMdKBrWuX5aAiG4ytL5ffWi0HBUjdleo=</NextToken>
    <MaxResults>1</MaxResults>
    <Consumers>
      <ConsumerTag>sgen-1</ConsumerTag>
    </Consumers>
  </Data>
</ListQueueConsumersResponse>
```

JSON 格式

```
{
  "RequestId": "4409B7D5-E4EC-4EB5-804A-385DCDFCD***",
  "Data": {
    "NextToken": "AAAAATzkBMTg4MDc3MDg2OTAyMzQyMAF0ZXN0AXF1ZXVlLXJhYmJpdC1zcHJpbmdib290LW
    FkdmFuY2U1ATc3NjMxNjIwATE1OTEzNTA5MDkzMdKBrWuX5aAiG4ytL5ffWi0HBUjdleo=",
    "MaxResults": 1,
    "Consumers": {
      "ConsumerTag": "sgen-1"
    }
  }
}
```

错误码

HttpCode	错误码	错误信息	描述
500	InternalServerError	The request processing has failed due to some unknown error.	请求因遭遇某些未知错误而失败。

访问[错误中心](#)查看更多错误码。

10.4. DeleteQueue

调用DeleteQueue删除Queue。

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	DeleteQueue	要执行的操作。取值： DeleteQueue 。
InstanceId	String	是	1880770869023** *	实例ID。
QueueName	String	是	DemoQueue	Queue名称。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	Test	Vhost名称。

返回数据

名称	类型	示例值	描述
RequestId	String	92385FD2-624A-48C9-8FB5-753F2AFA***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteQueue
&InstanceId=1880770869023***
&QueueName=DemoQueue
&RegionId=cn-hangzhou
&VirtualHost=Test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DeleteQueueResponse>
  <RequestId>92385FD2-624A-48C9-8FB5-753F2AFA***</RequestId>
</DeleteQueueResponse>
```

JSON 格式

```
{
  "RequestId": "92385FD2-624A-48C9-8FB5-753F2AFA***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

11.Exchange

11.1. CreateExchange

调用CreateExchange创建Exchange。

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	CreateExchange	要执行的操作。取值： CreateExchange 。
AutoDeleteState	Boolean	是	false	是否自动删除。取值： true：是。如果绑定到该Exchange的最后一个Queue解除绑定，该Exchange会自动删除。 false：否。如果绑定到该Exchange的最后一个Queue解除绑定，该Exchange不会自动删除。
ExchangeName	String	是	DemoExchange	Exchange名称。说明： - Exchange名称只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@），长度限制1~255 字符。 - Exchange创建后，名称不支持修改，只能删除后重建。

名称	类型	是否必选	示例值	描述
ExchangeType	String	是	DIRECT	Exchange类型。取值： • DIRECT：该类型路由规则会将消息路由到Binding Key与Routing Key完全匹配的Queue中。 • TOPIC：该类型与DIRECT类型相似，使用Routing Key模式匹配和字符串比较的方式将消息路由至绑定的Queue。 • FANOUT：该类型路由规则非常简单，会把所有发送到该Exchange的消息路由到所有与它绑定的Queue中，相当于广播功能。 • HEADERS：该类型与DIRECT类型相似。Headers Exchange使用Headers属性代替Routing Key进行路由匹配，在绑定Headers Exchange和Queue时，设置绑定属性的键值对；在向Headers Exchange发送消息时，设置消息的Headers属性键值对，使用消息Headers属性键值对和绑定属性键值对比较的方式将消息路由至绑定的Queue。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。
Internal	Boolean	是	false	是否为内部Exchange。取值： • false：否 • true：是
RegionId	String	是	cn-hangzhou	实例的地域ID。
VirtualHost	String	是	test	Exchange所在的Vhost名称。
AlternateExchange	String	否	DemoAE	备份Exchange。配置备份Exchange用于接收Exchange路由失败的消息。

返回数据

名称	类型	示例值	描述
RequestId	String	021788F6-E50C-4BD6-9F80-66B0A19A6***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=CreateExchange
&AutoDeleteState=false
&ExchangeName=DemoExchange
&ExchangeType=DIRECT
&InstanceId=amqp-cn-v0h1kb9nu***
&Internal=false
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<CreateExchangeResponse>
  <RequestId>021788F6-E50C-4BD6-9F80-66B0A19A6***</RequestId>
</CreateExchangeResponse>
```

JSON 格式

```
{
  "RequestId": "021788F6-E50C-4BD6-9F80-66B0A19A6***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

11.2. ListExchanges

调用ListExchanges查询Exchange。

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	ListExchanges	要执行的操作。取值： ListExchanges 。
InstanceId	String	是	amqp-cn-7pp2mwbc****	实例ID。

名称	类型	是否必选	示例值	描述
MaxResults	Integer	是	1	最大结果集。取值： 1 ~ 100。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。
NextToken	String	否	AAAANDQBYW1xc C1jbi03cHAYbXdiY 3AwMGEbDmhvc3 QBAXNkZndhYWJ hATE2NDkzM4 OTU5NDIB4o3z1p PwWzk4aYuiRffi8 R6-****	下一次查询的Token。取值： - 如果是第一次查询或没有下一次查询，则无需填写。 - 如果有下一次查询，则取值为上一次返回的 <code>NextToken</code> 值。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
Exchanges	Array of ExchangeVO		Exchange。
Attributes	Map	test	当前版本不支持此参数。 属性。
AutoDeleteState	Boolean	false	自动删除状态。
CreateTime	Long	1580886216000	创建时间戳，单位为毫秒。
ExchangeType	String	DIRECT	Exchange类型。
Name	String	amq.direct	Exchange名称。
VHostName	String	test	Vhost名称。
MaxResults	Integer	1	最大结果集。

名称	类型	示例值	描述
NextToken	String	AAAANDQBYW1xcC1j bi03cHAYbXdY3Aw MGEbDmhvc3QBAXN kZndhYWJhATE2NDk zMtM4OTU5NDIB4o 3z1pPwWzk4aYuiRff i8R6-****	表示当前调用返回的 <code>NextToken</code> 值，根据返回值判断是否有下一个查询开始的Token。 - 如果 <code>NextToken</code> 为空，则表示没有下一个查询开始的Token。 - 如果 <code>NextToken</code> 有返回值，则表示该返回值即为下一个查询开始的Token。
RequestId	String	FEBA5E0C-50D0- 4FA6-A794- 4901E5465***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListExchanges
&InstanceId=amqp-cn-7pp2mwbc****
&MaxResults=1
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListExchangesResponse>
  <RequestId>FEBA5E0C-50D0-4FA6-A794-4901E5465***</RequestId>
  <Data>
    <NextToken></NextToken>
    <MaxResults>1</MaxResults>
    <Exchanges>
      <VHostName>test</VHostName>
      <CreateTime>1580886216000</CreateTime>
      <AutoDeleteState>false</AutoDeleteState>
      <ExchangeType>DIRECT</ExchangeType>
      <Name>amq.direct</Name>
    </Exchanges>
  </Data>
</ListExchangesResponse>
```

JSON 格式

```
{
  "RequestId": "FEBA5E0C-50D0-4FA6-A794-4901E5465***",
  "Data": {
    "NextToken": "",
    "MaxResults": 1,
    "Exchanges": [
      {
        "VHostName": "test",
        "CreateTime": 1580886216000,
        "AutoDeleteState": false,
        "ExchangeType": "DIRECT",
        "Name": "amq.direct"
      }
    ]
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

11.3. DeleteExchange

调用DeleteExchange删除Exchange。

使用说明

- headers和x-jms-topic类型的Exchange不支持删除。
- Vhost下的3个内置Exchange（amq.direct、amq.topic和amq.fanout）不支持删除。

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	DeleteExchange	要执行的操作。取值： DeleteExchange 。
ExchangeName	String	是	DemoExchange	Exchange名称。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。

名称	类型	是否必选	示例值	描述
VirtualHost	String	是	test	Vhost名称。

返回数据

名称	类型	示例值	描述
RequestId	String	6961FFB8-6358-4EDC-9E3C-4A0C56CE6***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteExchange
&ExchangeName=DemoExchange
&InstanceId=amqp-cn-v0h1kb9nu***
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DeleteExchangeResponse>
  <RequestId>6961FFB8-6358-4EDC-9E3C-4A0C56CE6***</RequestId>
</DeleteExchangeResponse>
```

JSON 格式

```
{
  "RequestId": "6961FFB8-6358-4EDC-9E3C-4A0C56CE6***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.Binding

12.1. CreateBinding

调用CreateBinding为Exchange绑定Queue或者Exchange。

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	CreateBinding	要执行的操作。取值： CreateBinding 。
Argument	String	是	x-match:all;type:report;format:pdf	设置消息头属性键值对信息，消息头属性使用一个或多个键值对拼接而成。x-match属性必须配置，其余属性可自定义。x-match属性取值如下： all：默认值，所有消息头的键值对必须匹配。 any：至少一对消息头的键值对必须匹配。 属性之间使用半角分号（;）隔开，属性键与值之间使用半角冒号（:）区分。例如x-match:all;type:report;format:pdf 该参数适用于Headers Exchange，对其它类型的Exchange无效。其它类型的Exchange，此处可以填任意值。

名称	类型	是否必选	示例值	描述
BindingKey	String	是	.test	绑定键。 - 绑定源Exchange为非Topic类型： - 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、正斜线（/）、at符号（@）。 - 长度限制1~255字符。 - 绑定源Exchange为Topic类型： - 可以包含字母、数字、短划线（-）、下划线（_）、星号（*）、半角句号（.）、井号（#）、正斜线（/）、at符号（@）。 - 不能以半角句号（.）开头或结尾。对于井号（#）或星号（*），如果以其开头，则其后需有半角句号（.），如果以其结尾，则其前需有半角句号（.），如果既不是开头也不是结尾，则其前后均需有半角句号（.）。 - 长度限制1~255字符。
BindingType	String	是	0	绑定目标对象的类型。取值： 0：Queue 1：Exchange
DestinationName	String	是	DemoQueue	绑定目标名称。 此绑定目标需在控制台创建且所属Vhost与SourceExchange所属Vhost一致，即在VirtualHost中。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
SourceExchange	String	是	NormalEX	源Exchange名称，且此Exchange已在控制台创建。
VirtualHost	String	是	test	Vhost名称。 此Vhost已在控制台创建且是DestinationName和SourceExchange所属的Vhost。

返回数据

名称	类型	示例值	描述
RequestId	String	09768C14-E51C-4F4A-8077-30810032C***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=CreateBinding
&Argument=x-match:all;type:report;format:pdf
&BindingKey=.test
&BindingType=0
&DestinationName=DemoQueue
&InstanceId=amqp-cn-v0h1kb9nu***
&RegionId=cn-hangzhou
&SourceExchange=NormalEX
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<CreateBindingResponse>
  <RequestId>09768C14-E51C-4F4A-8077-30810032C***</RequestId>
</CreateBindingResponse>
```

JSON 格式

```
{
  "RequestId": "09768C14-E51C-4F4A-8077-30810032C***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.2. ListBindings

调用ListBindings查询绑定。

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	ListBindings	要执行的操作。取值： ListBindings 。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	1	本次读取的最大数据记录数量。取值： 1~100
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	Test	Vhost名称。
NextToken	String	否	caeba0bbb2be03 f84eb48b699f0a* ***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用和当返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
Bindings	Array of BindingDO		绑定。
Argument	String	all	x-match属性。取值： all：默认值，所有消息头的键值对必须匹配。 any：至少一对消息头的键值对必须匹配。 该参数仅适用于Headers类型的Exchange。

名称	类型	示例值	描述
BindingKey	String	amq.test	绑定键。 - 绑定源Exchange为非Topic类型： - 只能包含字母、数字、短划线 (-)、下划线 (_)、半角句号 (.)、正斜线 (/)、at符号 (@)。 - 长度限制1~255字符。 - 绑定源Exchange为Topic类型： - 可以包含字母、数字、短划线 (-)、下划线 (_)、星号 (*)、半角句号 (.)、井号 (#)、正斜线 (/)、at符号 (@)。 - 不能以半角句号 (.) 开头或结尾。对于井号 (#) 或星号 (*)，如果以其开头，则其后需有半角句号 (.)，如果以其结尾，则其前需有半角句号 (.)，如果既不是开头也不是结尾，则其前后均需有半角句号 (.)。 - 长度限制1~255字符。
BindingType	String	QUEUE	绑定目标对象的类型。取值： - QUEUE - EXCHANGE
DestinationName	String	QueueTest	绑定目标名称。
SourceExchange	String	test	源Exchange名称。
MaxResults	Integer	1	本次请求所返回的最大记录条数。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
RequestId	String	E0A71208-3E87-4732-81CC-B18E0B4B1***	请求ID。

示例

请求示例


```
http(s)://[Endpoint]/?Action=ListBindings
&InstanceId=1880770869023***
&MaxResults=1
&RegionId=cn-hangzhou
&VirtualHost=Test
&NextToken=caeba0bbb2be03f84eb48b699f0a****
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListBindingsResponse>
  <RequestId>E0A71208-3E87-4732-81CC-B18E0B4B1***</RequestId>
  <Data>
    <NextToken>,</NextToken>
    <MaxResults>1</MaxResults>
    <Bindings>
      <SourceExchange>test</SourceExchange>
      <Argument>all</Argument>
      <BindingType>QUEUE</BindingType>
      <BindingKey>amq.test</BindingKey>
      <DestinationName>QueueTest</DestinationName>
    </Bindings>
  </Data>
</ListBindingsResponse>
```

JSON 格式

```
{
  "RequestId": "E0A71208-3E87-4732-81CC-B18E0B4B1***",
  "Data": {
    "NextToken": ",",
    "MaxResults": "1",
    "Bindings": [
      {
        "SourceExchange": "test",
        "Argument": "all",
        "BindingType": "QUEUE",
        "BindingKey": "amq.test",
        "DestinationName": "QueueTest"
      }
    ]
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.3. ListDownStreamBindings

调用ListDownStreamBindings查询Exchange绑定。

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	ListDownStreamBindings	要执行的操作。取值： ListDownStreamBindings 。
ExchangeName	String	是	test	Exchange名称。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	1	最大结果集。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a* ***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Data	Struct		返回数据。
Bindings	Array of BindingVO		绑定。

名称	类型	示例值	描述
Argument	String	test	x-match属性。取值： all：默认值，所有消息头的键值对必须匹配。 any：至少一对消息头的键值对必须匹配。 该参数仅适用于Headers类型的Exchange，对其它类型的Exchange无效。
BindingKey	String	amq.test	绑定键。 - 绑定源Exchange为非Topic类型： - 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、正斜线（/）、at符号（@）。 - 长度限制1~255字符。 - 绑定源Exchange为Topic类型： - 可以包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@）。 - 不能以半角句号（.）开头或结尾。对于井号（#）或星号（*），如果以其开头，则其后需有半角句号（.），如果以其结尾，则其前需有半角句号（.），如果既不是开头也不是结尾，则其前后均需有半角句号（.）。 - 长度限制1~255字符。
BindingType	String	QUEUE	绑定目标对象的类型。取值： QUEUE EXCHANGE
DestinationName	String	QueueTest	绑定目标名称。
SourceExchange	String	test	源Exchange名称。
MaxResults	Integer	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
Message	String	operation success	返回信息。

名称	类型	示例值	描述
RequestId	String	9C1E0502-0790-4FDB-8C96-6D5C8D9B7***	请求ID。
Success	Boolean	true	是否成功。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListDownStreamBindings
&ExchangeName=test
&InstanceId=1880770869023***
&MaxResults=1
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<Message>operation success</Message>
<RequestId>9C1E0502-0790-4FDB-8C96-6D5C8D9B7***</RequestId>
<Data>
  <NextToken>,</NextToken>
  <MaxResults>1</MaxResults>
  <Bindings>
    <Argument>test</Argument>
    <SourceExchange>test</SourceExchange>
    <BindingKey>amq.test</BindingKey>
    <BindingType>QUEUE</BindingType>
    <DestinationName>QueueTest</DestinationName>
  </Bindings>
</Data>
<Code>200</Code>
<Success>true</Success>
```

JSON 格式

```
{
  "Message": "operation success",
  "RequestId": "9C1E0502-0790-4FDB-8C96-6D5C8D9B7***",
  "Data": {
    "NextToken": "",
    "MaxResults": 1,
    "Bindings": {
      "Argument": "test",
      "SourceExchange": "test",
      "BindingKey": "amq.test",
      "BindingType": "QUEUE",
      "DestinationName": "QueueTest"
    }
  },
  "Code": 200,
  "Success": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.4. ListExchangeUpStreamBindings

调用ListExchangeUpStreamBindings查询Exchange被绑定。

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	ListExchangeUpStreamBindings	要执行的操作。取值： ListExchangeUpStreamBindings。
ExchangeName	String	是	test	Exchange名称。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	1	最大结果集。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。

名称	类型	是否必选	示例值	描述
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Code	Integer	200	返回码。返回200代表成功。
Data	Struct		返回数据。
Bindings	Array of BindingVO		绑定。
Argument	String	all	x-match属性。取值： - all：默认值，所有消息头的键值对必须匹配。 - any：至少一对消息头的键值对必须匹配。 该参数仅适用于Headers类型的Exchange，对它其它类型的Exchange无效。
BindingKey	String	amq.dle.test	绑定键。 - 绑定源Exchange为非Topic类型： - 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、正斜线（/）、at符号（@）。 - 长度限制1~255字符。 - 绑定源Exchange为Topic类型： - 可以包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@）。 - 不能以半角句号（.）开头或结尾。对于井号（#）或星号（*），如果以其开头，则其后需有半角句号（.），如果以其结尾，则其前需有半角句号（.），如果既不是开头也不是结尾，则其前后均需有半角句号（.）。 - 长度限制1~255字符。

名称	类型	示例值	描述
BindingType	String	EXCHANGE	绑定目标对象的类型。取值： • QUEUE • EXCHANGE
DestinationName	String	test	目标名称。
SourceExchange	String	dle	源Exchange名称。
MaxResults	Integer	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
Message	String	operation success	返回信息。
RequestId	String	2DCCCE88-BC82-4A4F-AF5E-9A759672B***	请求ID。
Success	Boolean	true	是否成功。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListExchangeUpStreamBindings
&ExchangeName=test
&InstanceId=1880770869023***
&MaxResults=1
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListExchangeUpStreamBindingsResponse>
  <RequestId>2DCCCE88-BC82-4A4F-AF5E-9A759672B***</RequestId>
  <Message>operation success</Message>
  <Data>
    <NextToken>,</NextToken>
    <MaxResults>1</MaxResults>
    <Bindings>
      <SourceExchange>dle</SourceExchange>
      <Argument>all</Argument>
      <BindingType>EXCHANGE</BindingType>
      <BindingKey>amq.dle.test</BindingKey>
      <DestinationName>test</DestinationName>
    </Bindings>
  </Data>
  <Code>200</Code>
  <Success>true</Success>
</ListExchangeUpStreamBindingsResponse>
```

JSON 格式

```
{
  "RequestId": "2DCCCE88-BC82-4A4F-AF5E-9A759672B***",
  "Message": "operation success",
  "Data": {
    "NextToken": ",",
    "MaxResults": 1,
    "Bindings": {
      "SourceExchange": "dle",
      "Argument": "all",
      "BindingType": "EXCHANGE",
      "BindingKey": "amq.dle.test",
      "DestinationName": "test"
    }
  },
  "Code": 200,
  "Success": true
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.5. ListQueueUpStreamBindings

调用ListQueueUpStreamBindings查询Queue被绑定。

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	ListQueueUpStreamBindings	要执行的操作。取值： ListQueueUpStreamBindings 。
InstanceId	String	是	1880770869023** *	实例ID。
MaxResults	Integer	是	1	最大结果集。
QueueName	String	是	QueueTest	Queue名称。
RegionId	String	是	cn-hangzhou	地域ID。
VirtualHost	String	是	test	Vhost名称。
NextToken	String	否	caeba0bbb2be03f84eb48b699f0a* ***	用来标记当前查询结束的位置，以便下一次调用时作为参数传入，继续翻页。首次调用或返回最后一页时，取值为空字符串。

返回数据

名称	类型	示例值	描述
Data	Struct		返回数据。
Bindings	Array of BindingVO		绑定。
Argument	String	all	x-match属性。取值： all：默认值，所有消息头的键值对必须匹配。 any：至少一对消息头的键值对必须匹配。 该参数仅适用于Headers类型的Exchange，对其它类型的Exchange无效。

名称	类型	示例值	描述
BindingKey	String	amq.test	绑定键。 - 绑定源Exchange为非Topic类型： - 只能包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、正斜线（/）、at符号（@）。 - 长度限制1~255字符。 - 绑定源Exchange为Topic类型： - 可以包含字母、数字、短划线（-）、下划线（_）、半角句号（.）、井号（#）、正斜线（/）、at符号（@）。 - 不能以半角句号（.）开头或结尾。对于井号（#）或星号（*），如果以其开头，则其后需有半角句号（.），如果以其结尾，则其前需有半角句号（.），如果既不是开头也不是结尾，则其前后均需有半角句号（.）。 - 长度限制1~255字符。
BindingType	String	QUEUE	绑定目标对象的类型。取值： - QUEUE - EXCHANGE
DestinationName	String	QueueTest	绑定名称。
SourceExchange	String	test	源Exchange名称。
MaxResults	String	1	最大结果集。
NextToken	String	caebacccb2be03f84eb48b699f0a****	表示当前调用返回读取到的位置，空代表数据已经读取完毕。
RequestId	String	8BFB1C9D-08A2-4859-A47C-403C9EFA2***	请求ID。

示例

请求示例

```
http(s)://amqp-open.cn-hangzhou.aliyuncs.com/?Action=ListQueueUpStreamBindings
&InstanceId=1880770869023***
&MaxResults=1
&QueueName=QueueTest
&RegionId=cn-hangzhou
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ListQueueUpStreamBindingsResponse>
  <RequestId>8BFB1C9D-08A2-4859-A47C-403C9EFA2***</RequestId>
  <Data>
    <NextToken>,</NextToken>
    <MaxResults>1</MaxResults>
    <Bindings>
      <SourceExchange>test</SourceExchange>
      <Argument>all</Argument>
      <BindingType>QUEUE</BindingType>
      <BindingKey>amq.test</BindingKey>
      <DestinationName>QueueTest</DestinationName>
    </Bindings>
  </Data>
</ListQueueUpStreamBindingsResponse>
```

JSON 格式

```
{
  "RequestId": "8BFB1C9D-08A2-4859-A47C-403C9EFA2***",
  "Data": {
    "NextToken": ",",
    "MaxResults": 1,
    "Bindings": {
      "SourceExchange": "test",
      "Argument": "all",
      "BindingType": "QUEUE",
      "BindingKey": "amq.test",
      "DestinationName": "QueueTest"
    }
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

12.6. DeleteBinding

调用DeleteBinding解除源Exchange与目标Queue或者其他Exchange的绑定关系。

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	DeleteBinding	要执行的操作。取值： DeleteBinding 。
BindingKey	String	是	.test.	绑定键。
BindingType	String	是	QUEUE	解绑目标对象的类型。取值： • QUEUE • EXCHANGE
DestinationName	String	是	DemoQueue	解绑目标对象的名称。
InstanceId	String	是	amqp-cn-v0h1kb9nu***	实例ID。
RegionId	String	是	cn-hangzhou	地域ID。
SourceExchange	String	是	NormalEX	源Exchange名称。
VirtualHost	String	是	test	Vhost名称。

返回数据

名称	类型	示例值	描述
RequestId	String	5A60DBD2-2D9D-40EC-BEBF-C86E88817***	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DeleteBinding
&BindingKey=.test.
&BindingType=QUEUE
&DestinationName=DemoQueue
&InstanceId=amqp-cn-v0h1kb9nu***
&RegionId=cn-hangzhou
&SourceExchange=NormalEX
&VirtualHost=test
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DeleteBindingResponse>
  <RequestId>5A60DBD2-2D9D-40EC-BEBF-C86E88817***</RequestId>
</DeleteBindingResponse>
```

JSON 格式

```
{
  "RequestId": "5A60DBD2-2D9D-40EC-BEBF-C86E88817***"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.云监控资源报表接口

13.1. 云监控资源报表接口说明

您可使用云监控资源报表接口查询消息队列Rabbit MQ版的实例、Vhost、Queue和Exchange等资源的使用情况。

监控项

云监控资源报表支持以下消息队列Rabbit MQ版监控项指标。

资源类型	监控项	单位	Namespace	MetricName	Dimensions	Statistics
Instance	每秒消息流入数	个/秒	acs_amqp	InstanceTPSIn	userId、regionId、instanceId	Value
	每秒消息流出数	个/秒	acs_amqp	InstanceTPSOut	userId、regionId、instanceId	Value
	消费者数量	个	acs_amqp	InstanceConsumers	userId、regionId、instanceId	Value
	连接数量	个	acs_amqp	InstanceConnections	userId、regionId、instanceId	Value
	通道数量	个	acs_amqp	InstanceChannelsNew	userId、regionId、instanceId	Value
Vhost	每秒消息流入数	个/秒	acs_amqp	VHostTPSIn	userId、regionId、instanceId、vhostName	Value
	每秒消息流出数	个/秒	acs_amqp	VHostTPSOut	userId、regionId、instanceId、vhostName	Value
	消费者数量	个	acs_amqp	InstanceVhostConsumers	userId、regionId、instanceId、vhostName	Value
	连接数量	个	acs_amqp	VhostConnections	userId、regionId、instanceId、vhostName	Value

资源类型	监控项	单位	Namespace	MetricName	Dimensions	Statistics
	通道数量	个	acs_amqp	VhostChannels	userId、regionId、instanceId、vhostName	Value
Queue	每秒消息流入数	个/秒	acs_amqp	QueueTPSIn	userId、regionId、instanceId、vhostName、queueName	Value
	每秒消息流出数	个/秒	acs_amqp	QueueTPSOut	userId、regionId、instanceId、vhostName、queueName	Value
	消费者数量	个	acs_amqp	InstanceVhostQueueConsumers	userId、regionId、instanceId、vhostName、queueName	Value
	消息堆积量	个	acs_amqp	InstanceVhostQueueMessageAccum	userId、regionId、instanceId、vhostName、queueName	Maximum
Exchange	每秒消息流入数	个/秒	acs_amqp	ExchangeTPSIn	userId、regionId、instanceId、vhostName、exchangeName	Value
	每秒消息流出数	个/秒	acs_amqp	ExchangeTPSOut	userId、regionId、instanceId、vhostName、exchangeName	Value

API 概览

- DescribeMetricMetaList
- DescribeMetricLast

- [DescribeMetricList](#)
- [DescribeMetricData](#)
- [DescribeMetricTop](#)

调用 API

云监控资源报表接口的调用方法，请参见[调用方式](#)。

13.2. DescribeMetricMetaList

调用DescribeMetricMetaList接口查询云监控开放的时序类指标监控项描述。

通常配合查询监控数据接口DescribeMetricList和DescribeMetricLast一起使用，详情请参见[DescribeMetricList](#)和[DescribeMetricLast](#)。

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	DescribeMetricMetaList	系统规定参数，取值：DescribeMetricMetaList。
Namespace	String	否	acs_kvstore	指标所属的Namespace，用于区分每个产品配置。 详情请参见 云产品监控项 。
Labels	String	否	[{"name":"productCategory","value":"kvstore_old"}]	根据标签过滤。是一个JSON的字符串。 格式： [{"name":"标签名","value":"标签值"}, {"name":"标签名","value":"标签值"}] 。已有标签说明如下： • metricCategory：监控项分类的描述。 • alertEnable：是否需要报警。 • alertUnit：建议的报警单位。 • unitFactor：单位转换系数。 • minAlertPeriod：最小报警周期。 • productCategory：产品类型分类。
MetricName	String	否	CPUUtilization	监控项名称。详情请参见 云产品监控项 。
PageNumber	Integer	否	1	分页参数。默认值：1。

名称	类型	是否必选	示例值	描述
PageSize	Integer	否	30	每页最大数量。默认值：30。

返回数据

名称	类型	示例值	描述
Success	Boolean	true	操作是否成功。true表示成功，false表示失败。
Code	String	200	状态码。 说明 200表示成功。
RequestId	String	0CCE0AF0-053C-4B13-A583-DC9A85785D49	请求ID。
Message	String	The Request is not authorization.	错误信息。
Resources	Array		指定监控项的配置信息。
Resource			
Description	String	CPUUtilization	监控项描述。
Dimensions	String	instanceId	监控项的Dimensions。多个用英文逗号（,）分隔。

名称	类型	示例值	描述
Labels	String	<pre>[{"name\\":\\"alertUnit\\",\\"value\\":\\"Bytes\\"}, {"name\\":\\"minAlertPeriod\\",\\"value\\":\\"60\\"}, {"name\\":\\"metricCategory\\",\\"value\\":\\"instanceId\\"}, {"name\\":\\"instanceType\\",\\"value\\":\\"disaster\\"}, {"name\\":\\"is_alarm\\",\\"value\\":\\"true\\"}, {"name\\":\\"productCategory\\",\\"value\\":\\"kvstore_old\\"}]</pre>	监控项的标签，是一个或多个JSON字符串。格式：<code>[{"name": "标签名", "value": "标签的值"}]</code>，<code>name</code> 可以重复， 已有标签说明如下： <code>metricCategory</code>：监控项分类的描述。 <code>alertEnable</code>：是否需要报警。 <code>alertUnit</code>：建议的报警单位。 <code>unitFactor</code>：单位转换系数。 <code>minAlertPeriod</code>：最小报警周期。 <code>productCategory</code>：产品类型分类。
MetricName	String	CPUUtilization	监控项名称。
Namespace	String	acs_kvstore	用于区分产品类型。通常的命名方式为acs_产品名。
Periods	String	60,300	监控项的所有统计周期。多个用英文逗号（,）分隔。
Statistics	String	Average,Minimum,Maximum	统计方法。多个用英文逗号（,）分隔。
Unit	String	%	监控项的单位。
TotalCount	String	12	总记录条数。

示例

请求示例

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

正常返回示例

XML 格式

```
<DescribeMetricMetaListResponse>
  <TotalCount>1853</TotalCount>
  <RequestId>CDE9EAFD-D54E-4024-BBFC-B0AAC883143B</RequestId>
```

```

<Success>true</Success>
<Code>200</Code>
<Resources>
  <Resource>
    <Description>磁盘额定容量</Description>
    <Statistics>Average,Minimum,Maximum</Statistics>
    <MetricName>ads.diskSize</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"300"}, {"name":"alertDefault",
    "value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","valu
    e":"Mbytes"}, {"name":"productCategory","value":"ads"}, {"name":"is_alarm",
    "value":"true"}, {"name":"metricCategory","value":"workerId"}]</Labels>
    <Dimensions>userId,instanceId,tableSchema,workerId</Dimensions>
    <Namespace>acs_ads</Namespace>
    <Periods>300</Periods>
    <Unit>MB</Unit>
  </Resource>
  <Resource>
    <Description>磁盘已用容量</Description>
    <Statistics>Average,Minimum,Maximum</Statistics>
    <MetricName>ads.diskUsed</MetricName>
    <Labels>[{"name":"alertUnit","value":"Mbytes"}, {"name":"minAlertPeriod",
    "value":"300"}, {"name":"metricCategory","value":"workerId"}, {"name":"is_a
    larm","value":"true"}]</Labels>
    <Dimensions>userId,instanceId,tableSchema,workerId</Dimensions>
    <Namespace>acs_ads</Namespace>
    <Periods>300</Periods>
    <Unit>MB</Unit>
  </Resource>
  <Resource>
    <Description>磁盘使用率</Description>
    <Statistics>Average,Minimum,Maximum</Statistics>
    <MetricName>ads.diskUsedPercent</MetricName>
    <Labels>[{"name":"alertUnit","value":"%"}, {"name":"minAlertPeriod",""
    value":"300"}, {"name":"metricCategory","value":"workerId"}, {"name":"is_alarm",
    "value":"true"}]</Labels>
    <Dimensions>userId,instanceId,tableSchema,workerId</Dimensions>
    <Namespace>acs_ads</Namespace>
    <Periods>300</Periods>
    <Unit>%</Unit>
  </Resource>
  <Resource>
    <Description>queue的未消费的消息数目</Description>
    <Statistics>Maximum</Statistics>
    <MetricName>QueueMessageAccumulation</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault",
    "value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value
    ":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm",
    "value":"true"}, {"name":"metricCategory","value":"queue"}]</Labels>
    <Dimensions>userId,regionId,vhostName,queueName</Dimensions>
    <Namespace>acs_amqp</Namespace>
    <Periods>60,300</Periods>
    <Unit>count/min</Unit>
  </Resource>
  <Resource>

```

```

    <Description>queue每分钟消息生产量</Description>
    <Statistics>Value</Statistics>
    <MetricName>QueueMessageInput</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault",
    ",\n"value":"\n"}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","valu
    e":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm",
    ",\n"value":"true"}, {"name":"metricCategory","value":"queue"}]</Labels>
    <Dimensions>userId,regionId,vhostName,queueName</Dimensions>
    <Namespace>acs_amqp</Namespace>
    <Periods>60,300</Periods>
    <Unit>count/min</Unit>
</Resource>
<Resource>
    <Description>queue的未消费的消息数目</Description>
    <Statistics>Value</Statistics>
    <MetricName>QueueMessageOutput</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault",
    ",\n"value":"\n"}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value
    \":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm",
    ",\n"value":"true"}, {"name":"metricCategory","value":"queue"}]</Labels>
    <Dimensions>userId,regionId,vhostName,queueName</Dimensions>
    <Namespace>acs_amqp</Namespace>
    <Periods>60,300</Periods>
    <Unit>count/min</Unit>
</Resource>
<Resource>
    <Description>实例每分钟产生的消息数量</Description>
    <Statistics>Value</Statistics>
    <MetricName>VhostMessageInput</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault",
    ",\n"value":"\n"}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value
    \":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm",
    ",\n"value":"true"}, {"name":"metricCategory","value":"vhost"}]</Labels>
    <Dimensions>userId,regionId,vhostName</Dimensions>
    <Namespace>acs_amqp</Namespace>
    <Periods>60,300</Periods>
    <Unit>count/min</Unit>
</Resource>
<Resource>
    <Description>实例每分钟消费的消息数量</Description>
    <Statistics>Value</Statistics>
    <MetricName>VhostMessageOutput</MetricName>
    <Labels>[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault",
    ",\n"value":"\n"}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value
    \":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm",
    ",\n"value":"true"}, {"name":"metricCategory","value":"vhost"}]</Labels>
    <Dimensions>userId,regionId,vhostName</Dimensions>
    <Namespace>acs_amqp</Namespace>
    <Periods>60,300</Periods>
    <Unit>count/min</Unit>
</Resource>
<Resource>
    <Description></Description>
    <Statistics>Average</Statistics>
    <MetricName>Latency</MetricName>

```

```

<MetricName>Latency</MetricName>
  <Labels>[{"name":"alertUnit","value":"ms"}, {"name":"minAlertPeriod","value":"60"}, {"name":"metricCategory","value":"instanceId"}, {"name":"is_alarm","value":"true"}]</Labels>
  <Dimensions>userId,region,apiUid</Dimensions>
  <Namespace>acs_apigateway_dashboard</Namespace>
  <Periods>60,300,900</Periods>
  <Unit>ms</Unit>
</Resource>
<Resource>
  <Description>时间粒度内http返回码4XX占全部返回码的百分比</Description>
  <Statistics>Average,Minimum,Maximum</Statistics>
  <MetricName>code4xx</MetricName>
  <Labels>[{"name":"alertUnit","value":"%"}, {"name":"minAlertPeriod","value":"60"}, {"name":"metricCategory","value":"instanceId"}, {"name":"is_alarm","value":"true"}]</Labels>
  <Dimensions>userId,instanceId</Dimensions>
  <Namespace>acs_cdn</Namespace>
  <Periods>60,300</Periods>
  <Unit>%</Unit>
</Resource>
</Resources>
</DescribeMetricMetaListResponse>

```

JSON 格式

```

{
  "TotalCount": 1853,
  "RequestId": "CDE9EAFF-D54E-4024-BBFC-B0AAC883143B",
  "Success": true,
  "Code": 200,
  "Resources": {
    "Resource": [
      {
        "Description": "磁盘额定容量",
        "Statistics": "Average,Minimum,Maximum",
        "MetricName": "ads.diskSize",
        "Labels": "[{"name":"minAlertPeriod","value":"300"}, {"name":"alertDefault","value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value":"Mbytes"}, {"name":"productCategory","value":"ads"}, {"name":"is_alarm","value":"true"}, {"name":"metricCategory","value":"workerId"}]",
        "Dimensions": "userId,instanceId,tableSchema,workerId",
        "Namespace": "acs_ads",
        "Periods": "300",
        "Unit": "MB"
      },
      {
        "Description": "磁盘已用容量",
        "Statistics": "Average,Minimum,Maximum",
        "MetricName": "ads.diskUsed",
        "Labels": "[{"name":"alertUnit","value":"Mbytes"}, {"name":"minAlertPeriod","value":"300"}, {"name":"metricCategory","value":"workerId"}, {"name":"is_alarm","value":"true"}]",

```

```

        "Dimensions": "userId,instanceId,tableSchema,workerId",
        "Namespace": "acs_ads",
        "Periods": "300",
        "Unit": "MB"
    },
    {
        "Description": "磁盘使用率",
        "Statistics": "Average,Minimum,Maximum",
        "MetricName": "ads.diskUsedPercent",
        "Labels": "[{"name":"alertUnit","value":"%"}, {"name":"minAlertPeriod","value":"300"}, {"name":"metricCategory","value":"workerId"}, {"name":"is_alarm","value":"true"}]",
        "Dimensions": "userId,instanceId,tableSchema,workerId",
        "Namespace": "acs_ads",
        "Periods": "300",
        "Unit": "%"
    },
    {
        "Description": "queue的未消费的消息数目",
        "Statistics": "Maximum",
        "MetricName": "QueueMessageAccumulation",
        "Labels": "[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault","value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm","value":"true"}, {"name":"metricCategory","value":"queue"}]",
        "Dimensions": "userId,regionId,vhostName,queueName",
        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    },
    {
        "Description": "queue每分钟消息生产量",
        "Statistics": "Value",
        "MetricName": "QueueMessageInput",
        "Labels": "[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault","value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm","value":"true"}, {"name":"metricCategory","value":"queue"}]",
        "Dimensions": "userId,regionId,vhostName,queueName",
        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    },
    {
        "Description": "queue的未消费的消息数目",
        "Statistics": "Value",
        "MetricName": "QueueMessageOutput",
        "Labels": "[{"name":"minAlertPeriod","value":"60"}, {"name":"alertDefault","value":""}, {"name":"unitFactor","value":"1"}, {"name":"alertUnit","value":"count/min"}, {"name":"productCategory","value":"amqp"}, {"name":"is_alarm","value":"true"}, {"name":"metricCategory","value":"queue"}]",
        "Dimensions": "userId,regionId,vhostName,queueName",
        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    }
]

```

```

        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    },
    {
        "Description": "实例每分钟产生的消息数量",
        "Statistics": "Value",
        "MetricName": "VhostMessageInput",
        "Labels": "[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"aler\n\tDefault\",\"value\":\"\"},{\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\n\t\",\"value\":\"count/min\"},{\"name\":\"productCategory\",\"value\":\"amqp\"},{\"name\":\"is\n\t_alarm\",\"value\":\"true\"},{\"name\":\"metricCategory\",\"value\":\"vhost\"}]",
        "Dimensions": "userId,regionId,vhostName",
        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    },
    {
        "Description": "实例每分钟消费的消息数量",
        "Statistics": "Value",
        "MetricName": "VhostMessageOutput",
        "Labels": "[{\"name\":\"minAlertPeriod\",\"value\":\"60\"},{\"name\":\"aler\n\tDefault\",\"value\":\"\"},{\"name\":\"unitFactor\",\"value\":\"1\"},{\"name\":\"alertUnit\n\t\",\"value\":\"count/min\"},{\"name\":\"productCategory\",\"value\":\"amqp\"},{\"name\":\"is\n\t_alarm\",\"value\":\"true\"},{\"name\":\"metricCategory\",\"value\":\"vhost\"}]",
        "Dimensions": "userId,regionId,vhostName",
        "Namespace": "acs_amqp",
        "Periods": "60,300",
        "Unit": "count/min"
    },
    {
        "Description": "",
        "Statistics": "Average",
        "MetricName": "Latency",
        "Labels": "[{\"name\":\"alertUnit\",\"value\":\"ms\"},{\"name\":\"minAlertP\n\tperiod\",\"value\":\"60\"},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is\n\t_alarm\",\"value\":\"true\"}]",
        "Dimensions": "userId,region,apiUid",
        "Namespace": "acs_apigateway_dashboard",
        "Periods": "60,300,900",
        "Unit": "ms"
    },
    {
        "Description": "时间粒度内http返回码4XX占全部返回码的百分比",
        "Statistics": "Average,Minimum,Maximum",
        "MetricName": "code4xx",
        "Labels": "[{\"name\":\"alertUnit\",\"value\":\"%\"},{\"name\":\"minAlertPe\n\triod\",\"value\":\"60\"},{\"name\":\"metricCategory\",\"value\":\"instanceId\"},{\"name\":\"is\n\t_alarm\",\"value\":\"true\"}]",
        "Dimensions": "userId,instanceId",
        "Namespace": "acs_cdn",
        "Periods": "60,300",
        "Unit": "%"
    }
}

```

```
    }
  }
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.3. DescribeMetricLast

调用DescribeMetricLast接口查询指定监控对象的最新监控数据。

关于各个阿里云产品的Project、Metric、Period、Dimensions等参数的取值，请参见[DescribeMetricMetaList](#)或[预设监控项参考](#)。

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	DescribeMetricLast	系统规定参数，取值：DescribeMetricLast。
MetricName	String	是	CPUUtilization	监控项名称。
Namespace	String	是	acs_ecs_dashboa rd	产品的数据命名空间，用于区分不同的产品。 命名方式：acs_产品名。

名称	类型	是否必选	示例值	描述
Dimensions	String	否	<code>[{"instanceId": "XX-XXXX"}]</code>	维度map，用于查询指定资源的监控数据。 格式为 <code>key-value</code> 键值对形式的集合，常用的 <code>key-value</code> 集合为 <code>instanceId: XXXXXX</code>。 <code>key</code> 和 <code>value</code> 的长度为1~64个字节，超过64个字节时截取前64字节。<code>key</code> 和 <code>value</code> 的取值可包含字母、数字、点号（.）、短横线（-）、下划线（_）、正斜线（/）和反斜线（\）。 说明 <code>Dimensions</code> 传入时需要使用JSON字符串表示该map对象，必须按顺序传入。
EndTime	String	否	2019-01-31 10:10:00	结束时间。 格式为Unix时间戳，即从1970年1月1日开始所经过的秒数。 说明 只能查询270天以内的监控数据。
Express	String	否	<code>{"groupby": ["userId", "instanceId"]}</code>	对现有查询结果进行实时计算的表达式。
Length	String	否	1000	返回监控数据的每页大小，用于分页查询。 默认值为1000，即每页1000条监控数据。
NextToken	String	否	15761432850009dd70bb64cff1f0ff6c0b08ffff073be5fb1e785e2b020f7fed9b5e137bd810a6d6cff5ae***	分页游标的标识。 - 如果匹配查询条件的返回结果超过了分页大小，则会返回这个分页游标。 - 如果需要获取下一页数据，将返回的游标值作为请求参数即可，直到没有游标值的返回则表示已经获取了全部结果数据。

名称	类型	是否必选	示例值	描述
Period	String	否	60	时间间隔，通常为监控项的上报周期，单位为秒。  说明 - 如果不填写，则按照注册监控项时声明的上报周期来查询原始数据。 - 如果设置报警规则时设置了统计周期，则会按照此周期查询对应的统计数据。
StartTime	String	否	2019-01-31 10:00:00	开始时间。 格式为Unix时间戳，即从1970年1月1日开始所经过的秒数。  说明 只能查询270天以内且开始时间和结束时间不超过31天的监控数据。

返回数据

名称	类型	示例值	描述
RequestId	String	021472A6-25E3-4094-8D00-BA4B6A5486C3	请求ID。
Code	String	200	状态码。  说明 状态码为200表示成功。
Success	Boolean	true	请求是否成功。
Period	String	60	时间间隔，单位为秒。
NextToken	String	xxxxxx	分页游标标识。

名称	类型	示例值	描述
Datapoints	String	[[{"timestamp":1548777660000,"userId":"123","instanceId":"i-abc","Minimum":9.92,"Average":9.92,"Maximum":9.92}]]	监控数据列表。
Message	String	Success	错误信息。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeMetricLast
&MetricName=CPUUtilization
&Namespace=acs_ecs_dashboard
&<公共请求参数>
```

正常返回示例

XML 格式

```
<QueryMetricListResponse>
  <Period>60</Period>
  <Datapoints>
    <Datapoints>
      <timestamp>1490152860000</timestamp>
      <Maximum>100</Maximum>
      <userId> 123456789876****</userId>
      <Minimum>93.1</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.52</Average>
    </Datapoints>
    <Datapoints>
      <timestamp>1490152920000</timestamp>
      <Maximum>100</Maximum>
      <userId> 123456789876**** </userId>
      <Minimum>92.59</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.49</Average>
    </Datapoints>
    <Datapoints>
      <timestamp>1490152980000</timestamp>
      <Maximum>100</Maximum>
      <userId>123456789876****</userId>
      <Minimum>92.86</Minimum>
      <instanceId>i-abcdefgh12****</instanceId>
      <Average>99.44</Average>
    </Datapoints>
    <Datapoints>
```

```
<timestamp>1490153040000</timestamp>
<Maximum>100</Maximum>
<userId>123456789876****</userId>
<Minimum>91.43</Minimum>
<instanceId>i-abcdefgh12****</instanceId>
<Average>99.36</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153100000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.55</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.51</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153160000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.52</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153220000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.59</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.42</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153280000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.18</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.34</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153340000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.86</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.46</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153400000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.18</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.35</Average>
```

```
</Datapoints>
</Datapoints>
<RequestId>6661EC50-8625-4161-B349-E0DD59002AB7</RequestId>
<Success>true</Success>
<Code>200</Code>
</QueryMetricListResponse>
```

JSON 格式

```
{
  "Period": "60",
  "Datapoints": [
    {
      "timestamp": 1490152860000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 93.1,
      "Average": 99.52
    },
    {
      "timestamp": 1490152920000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 92.59,
      "Average": 99.49
    },
    {
      "timestamp": 1490152980000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 92.86,
      "Average": 99.44
    },
    {
      "timestamp": 1490153040000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 91.43,
      "Average": 99.36
    },
    {
      "timestamp": 1490153100000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 93.55,
      "Average": 99.51
    },
    {
      "timestamp": 1490153160000,
```

```
{
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":93.1,
  "Average":99.52
},
{
  "timestamp":1490153220000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.59,
  "Average":99.42
},
{
  "timestamp":1490153280000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.34
},
{
  "timestamp":1490153340000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.86,
  "Average":99.46
},
{
  "timestamp":1490153400000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.35
}
],
"RequestId":"6A5F022D-AC7C-460E-94AE-B9E75083D027",
"Success":true,
"Code":"200"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.4. DescribeMetricList

调用DescribeMetricList接口查询指定时间段内的云产品时序指标监控数据。

各云产品的Namespace、Project、Metric、Period、Dimensions等参数的取值，请参见[DescribeMetricMetaList](#)或[云监控主要监控项](#)。

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	DescribeMetricList	系统规定参数。取值：DescribeMetricList。
MetricName	String	是	cpu_idle	监控项名称。
Namespace	String	是	acs_ecs_dashboa rd	产品的数据命名空间，用于区分不同的产品。 命名方式：acs_产品名。
Dimensions	String	否	[{"instanceId": "i- abcdefgh12****"}]	维度map，用于查询指定资源的监控数据。 格式为 key-value 键值对形式的集合，常用的 key-value 集合为 instanceId: XXXXXX。 key 和 value 的长度为1~64个字节，超过64个字节时截取前64字节。key 和 value 的取值可包含字母、数字、点号（.）、短横线（-）、下划线（_）、正斜线（/）和反斜线（\）。 ❓ 说明 Dimensions 传入时需要使用JSON字符串表示该Map对象，必须按顺序传入。
EndTime	String	否	2019-01-30 00:10:00	结束时间。支持的格式： - Unix时间戳：从1970年1月1日开始所经过的秒数。 - Format格式：YYYY-MM-DDThh:mm:ssZ。
Express	String	否	{"groupby": ["userId","instanc eId"]}	对查询出的现有结果进行实时计算的表达式。目前仅支持 groupby（类似数据库的 groupby 语句）。

名称	类型	是否必选	示例值	描述
Length	String	否	1000	每页显示的记录条数，用于分页查询，默认值为1000。
NextToken	String	否	15761485350009 dd70bb64cff1f0f ff750b08ffff073b e5fb1e785e2b02 0f1a949d5ea14a ea7fed82f01dd8 7311	分页游标标识。  说明 如果不传入此参数表示获取第一页的数据，此参数只要返回就说明还有下一页，您可以将返回的NextToken作为参数再次请求获得下一页的数据，直到返回为null表示获取到了所有的数据。
Period	String	否	60	时间间隔，单位为秒。取值一般为60（1分钟）、300（5分钟）、900（15分钟）。  说明 请根据需要设置此参数。例如查询一天范围时设置Period为60，则返回1000条数据（实际存在1440，因为最大返回值不超过1000，则只返回前1000条）。如果使用Period为300，则返回288条数据。
StartTime	String	否	2019-01-30 00:00:00	开始时间。支持的格式： - Unix时间戳：从1970年1月1日开始所经过的秒数。 - Format格式：YYYY-MM-DDThh:mm:ssZ。  说明 开始和结束时间执行的是左开右闭的模式，startTime不能等于或大于endTime。

返回数据

名称	类型	示例值	描述
RequestId	String	3121AE7D-4AFF- 4C25-8F1D- C8226EBB1F42	请求ID。

名称	类型	示例值	描述
Code	String	200	状态码。 ? 说明 状态码为200表示成功。
Success	Boolean	true	请求是否成功。
NextToken	String	15761441850009dd70bb64cff1f0fff6d0b08ffff073be5fb1e785e2b020f7fed9b5e137bd810a6d6cff5aee816	分页游标标识。 ? 说明 如果不传入此参数表示获取第一页的数据，此参数只要返回就说明还有下一页，您可以将返回的NextToken作为参数再次请求获得下一页的数据，直到返回为null表示获取到了所有的数据。
Period	String	60	时间间隔，单位为秒。取值为60、300、900。
Datapoints	String	[{"timestamp":1548777660000,"userId":"123","instanceId":"i-abc","Minimum":9.92,"Average":9.92,"Maximum":9.92}]	监控数据列表。
Message	String	Success	错误信息。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeMetricList
&MetricName=cpu_idle
&Namespace=acs_ecs_dashboard
&<公共请求参数>
```

正常返回示例

XML 格式

```
<Period>60</Period>
<Datapoints>
  <timestamp>1490152860000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
```

```
</Datapoints>
<Datapoints>
  <timestamp>1490152920000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.59</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.49</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490152980000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.86</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.44</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153040000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.43</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.36</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153100000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.55</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.51</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153160000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.52</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153220000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.59</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.42</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153280000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
```

```

    <Minimum>91.18</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.34</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153340000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>92.86</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.46</Average>
</Datapoints>
<Datapoints>
    <timestamp>1490153400000</timestamp>
    <Maximum>100</Maximum>
    <userId>123456789876****</userId>
    <Minimum>91.18</Minimum>
    <instanceId>i-abcdefgh12****</instanceId>
    <Average>99.35</Average>
</Datapoints>
<RequestId>6A5F022D-AC7C-460E-94AE-B9E75083D027</RequestId>
<Success>true</Success>
<Code>200</Code>

```

JSON 格式

```

{
  "Period": "60",
  "Datapoints": [
    {
      "timestamp": 1490152860000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 93.1,
      "Average": 99.52
    },
    {
      "timestamp": 1490152920000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 92.59,
      "Average": 99.49
    },
    {
      "timestamp": 1490152980000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 92.86,
      "Average": 99.44
    },
    {

```

```
{
  "timestamp":1490153040000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.43,
  "Average":99.36
},
{
  "timestamp":1490153100000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":93.55,
  "Average":99.51
},
{
  "timestamp":1490153160000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":93.1,
  "Average":99.52
},
{
  "timestamp":1490153220000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.59,
  "Average":99.42
},
{
  "timestamp":1490153280000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.34
},
{
  "timestamp":1490153340000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.86,
  "Average":99.46
},
{
  "timestamp":1490153400000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.35
}
```

```
average: 100.00
    }
  ],
  "RequestId": "6A5F022D-AC7C-460E-94AE-B9E75083D027",
  "Success": true,
  "Code": "200"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.5. DescribeMetricData

调用DescribeMetricData接口查询指定时间段内的云产品时序指标监控数据。

各云产品的Namespace、Project、Metric、Period、Dimensions等参数的取值，请参见[DescribeMetricMetaList](#)或[云监控主要监控项](#)。

 **说明** 该参数与 `DescribeMetricList` 不同，这个接口带有统计功能，即Dimension={"userId":"xxx"}，把该用户下的所有数据进行聚合计算。

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	DescribeMetricData	系统规定参数。取值：DescribeMetricData。
MetricName	String	是	cpu_idle	监控项名称。
Namespace	String	是	acs_ecs_dashboa rd	产品的数据命名空间，用于区分不同的产品。 命名方式：acs_产品名。

名称	类型	是否必选	示例值	描述
Dimensions	String	否	<code>[{"instanceId": "i-abcdefgh12****"}]</code>	维度map，用于查询指定资源的监控数据。 格式为 <code>key-value</code> 键值对形式的集合，常用的 <code>key-value</code> 集合为 <code>instanceId: XXXXXX</code>。 <code>key</code> 和 <code>value</code> 的长度为1~64个字节，超过64个字节时截取前64字节。<code>key</code> 和 <code>value</code> 的取值可包含字母、数字、点号（.）、短横线（-）、下划线（_）、正斜线（/）和反斜线（\）。  说明 <code>Dimensions</code> 传入时需要使用JSON字符串表示该Map对象，必须按顺序传入。
EndTime	String	否	2019-01-30 00:10:00	结束时间。支持的格式： - Unix时间戳：从1970年1月1日开始所经过的秒数。 - Format格式：YYYY-MM-DDThh:mm:ssZ。
Express	String	否	<code>"{"groupby": ["userId","instanceId"]}"</code>	对查询出的现有结果进行实时计算的表达式。目前仅支持 <code>groupby</code> （类似数据库的 <code>groupby</code> 语句）。
Length	String	否	1000	每页显示的记录条数，用于分页查询，默认值为1000。
Period	String	否	60	时间间隔，单位为秒。取值一般为60（1分钟）、300（5分钟）、900（15分钟）。  说明 请根据需要设置此参数。例如查询一天范围时设置Period为60，则返回1000条数据（实际存在1440，因为最大返回值不超过1000，则只返回前1000条）。如果使用Period为300，则返回288条数据。

名称	类型	是否必选	示例值	描述
StartTime	String	否	2019-01-30 00:00:00	开始时间。支持的格式： • Unix时间戳：从1970年1月1日开始所经过的秒数。 • Format格式：YYYY-MM-DDThh:mm:ssZ。 ❓ 说明 开始和结束时间执行的是左开右闭的模式，startTime不能等于或大于endTime。

返回数据

名称	类型	示例值	描述
Code	String	200	状态码。 ❓ 说明 状态码为200表示成功。
Datapoints	String	[{"timestamp":1548777660000,"userId":"123****","instanceId":"i-abc****","Minimum":9.92,"Average":9.92,"Maximum":9.92}]	监控数据列表。
Message	String	Success	错误信息。
Period	String	60	时间间隔，单位为秒。取值为60、300、900。
RequestId	String	6A5F022D-AC7C-460E-94AE-B9E75083D027	请求ID。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeMetricData
&MetricName=cpu_idle
&Namespace=acs_ecs_dashboard
&<公共请求参数>
```

正常返回示例

XML 格式

```
<Period>60</Period>
<Datapoints>
  <timestamp>1490152860000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.52</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490152920000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.59</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.49</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490152980000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.86</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.44</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153040000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.43</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.36</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153100000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.55</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.51</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153160000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>93.1</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.52</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153220000</timestamp>
  <Maximum>100</Maximum>
```



```
<userId>123456789876****</userId>
<Minimum>92.59</Minimum>
<instanceId>i-abcdefgh12****</instanceId>
<Average>99.42</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153280000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.18</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.34</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153340000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>92.86</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.46</Average>
</Datapoints>
<Datapoints>
  <timestamp>1490153400000</timestamp>
  <Maximum>100</Maximum>
  <userId>123456789876****</userId>
  <Minimum>91.18</Minimum>
  <instanceId>i-abcdefgh12****</instanceId>
  <Average>99.35</Average>
</Datapoints>
<RequestId>6A5F022D-AC7C-460E-94AE-B9E75083D027</RequestId>
<Success>true</Success>
<Code>200</Code>
```

JSON 格式

```
{
  "Period": "60",
  "Datapoints": [
    {
      "timestamp": 1490152860000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 93.1,
      "Average": 99.52
    },
    {
      "timestamp": 1490152920000,
      "Maximum": 100,
      "userId": "123456789876****",
      "instanceId": "i-abcdefgh12****",
      "Minimum": 92.59,
      "Average": 99.49
    },
  ],
}
```

```
{
  "timestamp":1490152980000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.86,
  "Average":99.44
},
{
  "timestamp":1490153040000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.43,
  "Average":99.36
},
{
  "timestamp":1490153100000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":93.55,
  "Average":99.51
},
{
  "timestamp":1490153160000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":93.1,
  "Average":99.52
},
{
  "timestamp":1490153220000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.59,
  "Average":99.42
},
{
  "timestamp":1490153280000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":91.18,
  "Average":99.34
},
{
  "timestamp":1490153340000,
  "Maximum":100,
  "userId":"123456789876****",
  "instanceId":"i-abcdefgh12****",
  "Minimum":92.86,
```

```
    "Average":99.46
  },
  {
    "timestamp":1490153400000,
    "Maximum":100,
    "userId":"123456789876****",
    "instanceId":"i-abcdefgh12****",
    "Minimum":91.18,
    "Average":99.35
  }
],
"RequestId":"6A5F022D-AC7C-460E-94AE-B9E75083D027",
"Success":true,
"Code":"200"
}
```

错误码

访问[错误中心](#)查看更多错误码。

13.6. DescribeMetricTop

调用DescribeMetricTop接口查询指定时间段内排序后的云产品时序指标监控数据。

各云产品的Namespace、Orderby、Project、Metric、Period、Dimensions等参数的取值，请参见[DescribeMetricMetaList](#)或[云监控主要监控项](#)。

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	DescribeMetricTop	系统规定参数。取值：DescribeMetricTop。
MetricName	String	是	cpu_idle	监控项名称。
Namespace	String	是	acs_ecs_dashboa rd	产品的数据命名空间，用于区分不同的产品。 命名方式：acs_产品名。
Orderby	String	是	Average	排序字段，即按该排序字段进行排序，此参数为必填项。

名称	类型	是否必选	示例值	描述
Dimensions	String	否	<code>[{"instanceId": "i-abcdefgh12****"}]</code>	维度map，用于查询指定资源的监控数据。 格式为 <code>key-value</code> 键值对形式的集合，常用的 <code>key-value</code> 集合为 <code>instanceId: XXXXXX</code>。 <code>key</code> 和 <code>value</code> 的长度为1~64个字节，超过64个字节时截取前64字节。<code>key</code> 和 <code>value</code> 的取值可包含字母、数字、点号（.）、短横线（-）、下划线（_）、正斜线（/）和反斜线（\）。  说明 <code>Dimensions</code> 传入时需要使用JSON字符串表示该Map对象，必须按顺序传入。
EndTime	String	否	2019-01-30 00:10:00	结束时间。支持的格式： - Unix时间戳：从1970年1月1日开始所经过的秒数。 - Format格式：YYYY-MM-DDThh:mm:ssZ。
Express	String	否	<code>{"groupby": ["userId", "instanceId"]}</code>	对查询出的现有结果进行实时计算的表达式。目前仅支持 <code>groupby</code>（类似数据库的 <code>groupby</code> 语句）。
Length	String	否	1000	每页显示的记录条数，用于分页查询，默认值为1000。
OrderDesc	String	否	False	排序方式。取值： - True：由小到大排序 - False：由大到小排序

名称	类型	是否必选	示例值	描述
Period	String	否	60	时间间隔，单位为秒。取值一般为60（1分钟）、300（5分钟）、900（15分钟）。  说明 请根据需要设置此参数。例如查询一天范围时设置Period为60，则返回1000条数据（实际存在1440，因为最大返回值不超过1000，则只返回前1000条）。如果使用Period为300，则返回288条数据。
StartTime	String	否	2019-01-30 00:00:00	开始时间。支持的格式： • Unix时间戳：从1970年1月1日开始所经过的秒数。 • Format格式：YYYY-MM-DDThh:mm:ssZ。  说明 开始和结束时间执行的是左开右闭的模式，startTime不能等于或大于endTime。

返回数据

名称	类型	示例值	描述
RequestId	String	3121AE7D-4AFF-4C25-8F1D-C8226EBB1F42	请求ID。
Code	String	200	状态码。  说明 状态码为200表示成功。
Datapoints	String	[{"timestamp":1548777660000,"userId":"123","instanceId":"i-abc","Minimum":9.92,"Average":9.92,"Maximum":9.92}]	监控数据列表。
Period	String	60	时间间隔，单位为秒。取值为60、300、900。
Message	String	Success	错误信息。

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeMetricTop
&MetricName=cpu_idle
&Namespace=acs_ecs_dashboard
&Orderby=Average
&<公共请求参数>
```

正常返回示例

XML 格式

```
<Period>60</Period>
<Datapoints>
  <order>1</order>
  <timestamp>1551687360000</timestamp>
  <userId>12345****</userId>
  <instanceId>i-2zechst1****</instanceId>
  <Maximum>16.41</Maximum>
  <Minimum>4.66</Minimum>
  <Average>7.74</Average>
  <_count>1</_count>
</Datapoints>
<Datapoints>
  <order>2</order>
  <timestamp>1551687360000</timestamp>
  <userId>12345****</userId>
  <instanceId>i-2zefxdy2****</instanceId>
  <Maximum>15.74</Maximum>
  <Minimum>5.03</Minimum>
  <Average>7.14</Average>
  <_count>1</_count>
</Datapoints>
<RequestId>1F68A4E8-4488-48E7-9189-3E1F5165E64E</RequestId>
<Code>200</Code>
```

JSON 格式

```
{
  "Period": "60",
  "Datapoints": [
    {
      "timestamp": 1551687360000,
      "order": 1,
      "_count": 1,
      "Maximum": 16.41,
      "userId": "12345****",
      "Minimum": 4.66,
      "instanceId": "i-2zeehst1****",
      "Average": 7.74
    },
    {
      "timestamp": 1551687360000,
      "order": 2,
      "_count": 1,
      "Maximum": 15.74,
      "userId": "12345****",
      "Minimum": 5.03,
      "instanceId": "i-2zefxdy2****",
      "Average": 7.14
    }
  ],
  "RequestId": "1F68A4E8-4488-48E7-9189-3E1F5165E64E",
  "Code": "200"
}
```

错误码

访问[错误中心](#)查看更多错误码。