

ALIBABA CLOUD

# Alibaba Cloud

消息服务MNS  
API参考

文档版本：20201106

 阿里云

## 法律声明

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

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

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	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.基本概念 ----- 07
  - 1.1. Account ----- 07
  - 1.2. Queue ----- 07
  - 1.3. QueueMessage ----- 09
  - 1.4. Topic ----- 10
  - 1.5. TopicMessage ----- 11
  - 1.6. Subscription ----- 12
  - 1.7. Endpoint ----- 13
  - 1.8. NotifyStrategy ----- 13
  - 1.9. NotifyContentFormat ----- 14
- 2.调用方式 ----- 16
  - 2.1. 请求结构 ----- 16
  - 2.2. 公共参数 ----- 17
  - 2.3. 返回结果 ----- 18
  - 2.4. 请求签名机制 ----- 19
  - 2.5. Endpoint签名认证 ----- 22
- 3.队列接口规范 ----- 27
  - 3.1. RESTful API概览 ----- 27
  - 3.2. 队列管理 ----- 27
    - 3.2.1. CreateQueue ----- 28
    - 3.2.2. SetQueueAttributes ----- 30
    - 3.2.3. GetQueueAttributes ----- 32
    - 3.2.4. DeleteQueue ----- 35
    - 3.2.5. ListQueue ----- 36
  - 3.3. 消息管理 ----- 38
    - 3.3.1. SendMessage ----- 38

---

3.3.2. BatchSendMessage	41
3.3.3. ReceiveMessage	45
3.3.4. BatchReceiveMessage	48
3.3.5. DeleteMessage	50
3.3.6. BatchDeleteMessage	52
3.3.7. PeekMessage	54
3.3.8. BatchPeekMessage	57
3.3.9. ChangeMessageVisibility	59
4.主题接口规范	62
4.1. RESTful API概览	62
4.2. 主题管理	63
4.2.1. CreateTopic	63
4.2.2. SetTopicAttributes	65
4.2.3. GetTopicAttributes	66
4.2.4. DeleteTopic	68
4.2.5. ListTopic	69
4.3. 订阅管理	71
4.3.1. Subscribe	71
4.3.2. SetSubscriptionAttributes	74
4.3.3. GetSubscriptionAttributes	75
4.3.4. Unsubscribe	77
4.3.5. ListSubscriptionByTopic	78
4.4. 消息管理	80
4.4.1. PublishMessage	80
4.5. 通知管理	84
4.5.1. HttpEndpoint	84
5.错误响应	88
5.1. 错误响应格式	88

---

---

5.2. 错误码	88
6.API问题	92
6.1. API使用说明	92
6.2. 为什么我的签名总是没算对?	92

# 1. 基本概念

## 1.1. Account

用户的阿里云账号ID，可承担不同角色。

### *Account*

阿里云的注册用户，在MNS中以阿里云注册帐号的ID表示。

## 队列模型

### *队列所有者*

开通MNS服务的Account通过>CreateQueue接口创建一个消息队列，这个Account就是这个队列的所有者，队列所有者拥有此队列的所有操作权限。队列所有者对应的Account ID在阿里云官网可以查看。

### *生产者*

向MNS的消息队列发送消息的角色。

### *消费者*

从MNS的消息队列获取消息的角色。

## 主题模型

### *主题所有者*

开通MNS服务的Account通过>CreateTopic接口创建一个主题，这个Account就是该主题的所有者，主题所有者拥有该主题的所有操作权限。

### *发布者*

向MNS的主题发布消息的角色。

### *订阅者*

从MNS的主题接收消息的角色。

## 1.2. Queue

队列是消息存储的目的地，队列可以分成普通队列和延时队列。如果发送消息时不指定消息延时，被发送到普通队列的消息立刻可以被消费，而发送到延时队列需要经过设定的延时时间后才能被消费。

### *QueueURL*

- 格式：`http://$AccountId.mns.<Region>.aliyuncs.com/queues/$QueueName`。
- `mns.<Region>.aliyuncs.com`：MNS访问域名，Region是MNS部署的地域，您可以根据应用需要选择不同的地域。
- `AccountId`：队列所有者的账号ID。
- `QueueName`：队列名称，同一个AccountId在同地域中的队列名不能重名。

## 队列属性

### *DelaySeconds*

- 消息延迟时间，单位：秒。
- 取值范围：0 ~ 604800秒（7天）。
- 当该属性大于0时，发送到队列的所有消息是延迟消息，经过该属性指定的时间后变为可消费消息。
- 如果发送消息请求指定了DelaySeconds，以请求指定的DelaySeconds为准。

### *MaximumMessageSize*

- 消息体最大长度，单位：Byte。
- 取值范围：1024 ~ 65536 Byte（64 KB）。
- 该属性用于限制发送到队列的消息体的最大长度，超过该长度发送消息失败。

### *MessageRetentionPeriod*

- 消息最长保留时间，单位：秒。
- 取值范围：60 ~ 604800秒（7天）。
- 发送到队列的消息最长保留该时长，超过指定时间，无论消息是否被消费都将被删除。

### *VisibilityTimeout*

- 消息被receive后的隐藏时长，单位：秒。
- 取值范围：1 ~ 43200秒（12小时）。
- 消息被receive后，在该属性指定的这段时间内消息处于隐藏状态。在这段时间内，可以删除消息或者修改消息隐藏时长；超过这段时间，消息可以再次被receive。

### *PollingWaitSeconds*

- (batch)receive message请求最长等待时间，单位：秒。
- 取值范围：0 ~ 30秒。
- 当队列中没有消息时，(batch)receive message请求将运行在MNS服务器端。在该属性指定的时间范围内，如果有消息发送到队列中，立即返回消息给用户；如果超过该时间，仍然没有消息，返回MessageNotExist。
- 如果(batch)receive message请求指定了wait seconds，以请求指定的时间为准。

### *LoggingEnabled*

- 是否开启日志管理功能。
- 取值范围：True/False。
- 当该属性为True时，MNS将收集队列的消息操作日志按照指定的方式进行推送，日志管理请参见[日志管理概述](#)。

### *CreateTime*

队列创建时间，从1970-1-1 00:00:00到现在的秒值。

### *LastModifyTime*

修改队列属性信息的最近时间，从1970-1-1 00:00:00到现在的秒值。

### *ActiveMessages*

队列中处于Active状态的消息总数，为近似值。



### *InactiveMessages*

队列中处于Inactive状态的消息总数，为近似值。

### *DelayMessages*

队列中处于Delayed状态的消息总数，为近似值。

## 1.3. QueueMessage

本文介绍队列消息的消息属性、消息类型、消息状态。

### 队列消息

用户发送消息到队列。发送消息时可以指定消息体、消息延时时长、消息优先级。

### 消息属性

#### *MessageId*

- 消息编号。
- (batch) send message、receive message和peek message操作返回该属性。
- 一个队列中每个消息都有唯一的MessageId。
- 消息发送到队列中，MNS会生成一个MessageId，该编号一旦生成就不会变化，可以用来做数据校对。

#### *Next VisibleTime*

- 消息下次可被消费的时间，从1970年1月1日00:00:00 000开始的毫秒数。
- (batch) receive message和change message visibility操作返回该属性。

#### *Receipt Handle*

- 消息临时句柄。
- (batch) receive message和change message visibility操作返回该属性。
- 该句柄用于删除和修改处于Inactive状态的消息，NextVisibleTime之前有效，超过该时间使用句柄MNS会提示MessageNotExist。
- 消息临时句柄只能使用一次，如果该句柄标识的消息状态发生改变，该句柄就会失效。

#### *MessageBody*

- 消息正文。
- (batch) receive message和peek message操作返回该属性。

#### *MessageBodyMD5*

- 消息正文的MD5值。
- (batch) send message、receive message和peek message操作返回该属性。

#### *EnqueueTime*

- 消息发送到队列的时间，从1970-1-1 00:00:00 000开始的毫秒数。
- (batch) receive message和peek message操作返回该属性。

### *FirstDequeueTime*

- 消息第一次被消费的时间，从1970-1-1 00:00:00 000开始的毫秒数。
- (batch) receive message和peek message操作返回该属性。
- 如果消息从未被消费过，该属性与EnqueueTime相同。

### *DequeueCount*

- 消息总共被消费的次数，即被receive的次数。
- (batch) receive message和peek message操作返回该属性。

### *Priority*

- 消息的优先级权值，取值范围是1~16，其中1是最高优先级。
- (batch) receive message和peek message操作返回该属性。
- 如果队列中有不同优先级的消息，优先级越高的消息越容易更早被消费，即被ReceiveMessage操作取出。
- MNS会尽量让高优先级的消息先出队列，正是因为分布式消息队列的一些特性不能百分之百保证高优先级的消息先被消费。

## 消息类型

### *延时消息*

- 这类消息发送到队列后，需要经过DelaySeconds指定的时间后才可被消费。
- 发送(batch)send message请求时指定或者配置队列的DelaySeconds属性。

## 消息状态

- 普通消息被发送到普通队列时，初始状态是Active，当其被取走后在VisibilityTimeout的时间内状态为Inactive。若超过VisibilityTimeout时间后消息未被删除，消息会重新变成Active状态；如果在VisibilityTimeout时间内被删除，消息状态变为Deleted。
- 普通消息发送到延时队列时，消息初始状态Delayed，经过延时队列的DelaySeconds属性值设定的时间后，消息状态变成Active。
- 延时消息发送到消息队列（普通队列或者延时队列），消息的初始状态为Delayed，经过消息的DelaySeconds属性指定的时间后消息状态变成Active。
- 消息的最长存活时间由创建队列时指定的MessageRetentionPeriod属性值决定，超过此时间后消息状态会变成Expired，将被垃圾回收器回收。
- 消费者只能取到处于Active状态的消息。

# 1.4. Topic

主题是发布消息的目的地，发布者可以通过PublishMessage接口向主题发布消息。

### *TopicURL*

标识主题的URL，格式：`http://$AccountId.mns.<Region>.aliyuncs.com/topics/$TopicName`。

- `mns.<Region>.aliyuncs.com`：MNS访问域名。

Region是MNS部署的地域，您可以根据应用需要选择不同的地域，具体各地域的访问域名可以在控制台获取Endpoint查看。

- **AccountId** : 主题所有者的账号ID。

实际调用MNS的API时请换成实际的账号ID, 查看方法请参见[快速入门](#)。

- **TopicName** : 主题的名称。

MNS使用者在实际调用MNS的API时替换成实际的主题名称。同一个AccountId在同一个地域中的主题名称不能重名。

## 主题属性

### *MaximumMessageSize*

- 消息体的最大长度, 单位为Byte。
- 取值范围: 1024 Byte ~ 65536 Byte (64 KB) 。
- 该属性用于限制发送到主题的消息体最大长度, 超过该长度发布消息将失败。

### *LoggingEnabled*

- 是否开启日志管理功能。
- 取值范围: True或者False。
- 当该属性为True时, MNS将收集主题的消息操作日志推送到指定的地方, 日志管理请参见[日志管理概述](#)。

### *CreateTime*

主题的创建时间, 从1970-1-1 00:00:00到现在的秒值。

### *LastModifyTime*

修改主题属性信息的最近时间, 从1970-1-1 00:00:00到现在的秒值。

### *MessageRetentionPeriod*

- 消息在主题中的最长保留时间, 单位为秒。
- 从发送消息到该主题开始经过此参数指定的时间后, 不论消息是否被成功推送给用户都将被删除。

### *MessageCount*

- 该主题中消息数目。
- 包含已经被推送给用户的消息, 不包含过期被回收的消息。

## 1.5. TopicMessage

用户发布消息到主题时, 可以指定消息体、消息标签和消息额外属性。主题消息的最长保留时间为1天, 超过这个时间, 不论消息是否成功推送给用户都将被垃圾回收器回收。

## 消息属性

### *MessageId*

- 消息的编号。
- 一个主题中每个消息都有唯一的MessageId。

### *Message*

消息正文。

### *MessageMD5*

消息正文的MD5值。

### *MessageTag*

- 消息的标签。
- 当发布消息时指定了消息标签，MNS将只推送消息给接收这类标签消息的订阅。

### *PublishTime*

消息的发布时间，从1970-1-1 00:00:00 000到消息发布时的毫秒值。

## 1.6. Subscription

Subscription描述一个订阅关系，包括被订阅的主题和接收消息的Endpoint；订阅者通过Subscribe接口创建订阅，MNS根据订阅的描述给用户推送消息，推送的起点是订阅时Topic中最新的消息；订阅创建成功后，MNS会向指定的Endpoint推送消息。如果Endpoint不可用，MNS会按照指定的重试策略进行重试，重试失败后，会丢弃消息。

### *SubscriptionURL*

标识Subscription的URL格式：`http://$AccountId.mns.`

`<Region>.aliyuncs.com/topic/$TopicName/subscriptions/$SubscriptionName`。

- `mns.<Region>.aliyuncs.com`：MNS访问域名。

Region是MNS部署的地域，您可以根据应用需要选择不同的区域，具体各地域的访问域名可以在控制台获取Endpoint查看。

- `AccountId`：主题所有者的账号ID。

实际调用MNS的API时请替换成实际的账号ID。

- `TopicName`：主题的名称。

实际调用MNS的API时请替换成实际的主题名称。

- `SubscriptionName`：Subscription的名称。

实际调用MNS的API时请替换成实际Subscription的名称。同一个订阅者订阅相同的主题时，SubscriptionName不能重复。

## 订阅属性

### *Endpoint*

- 接收消息的Endpoint。
- 支持类型：HttpEndpoint、QueueEndpoint和MailEndpoint。

### *FilterTag*

- 过滤消息的标签。
- 类型：字符串。

- 指定了该属性后，MNS仅推送消息标签与该属性一致的消息到对应的Endpoint。

### *NotifyStrategy*

- 消息推送出现错误时的重试策略。
- 取值范围：BACKOFF\_RETRY或者EXPONENTIAL\_DECAY\_RETRY。
- 当推送消息失败时，MNS将按照该属性指定的重试策略进行重试。超过重试策略的次数后，MNS将丢弃这条消息，继续推送下一条消息。

具体的重试策略请参见[NotifyStrategy](#)。

### *NotifyContentFormat*

- 推送给Endpoint的消息格式。
- 取值范围：XML或者SIMPLIFIED。

具体推送消息的格式请参见[NotifyContentFormat](#)。

### *CreateTime*

订阅的创建时间，从1970-1-1 00:00:00到现在的秒值。

### *LastModifyTime*

修改订阅属性信息的最近时间，从1970-1-1 00:00:00到现在的秒值。

## 1.7. Endpoint

Endpoint是用户订阅主题时，指定接收消息的终端地址。当有消息发布到主题时，MNS会主动将消息推送到对应的Endpoint。多个Subscription可以指定同一个Endpoint。

### *HttpEndpoint*

- HTTP格式的Endpoint。
- 格式：`http://$company.com[/uri]`，支持多级URI。
- 限制：Endpoint的URI不能以“mns-reserved-”开头。
- MNS将发送HTTP请求到指定的Endpoint，用户在Endpoint端处理请求即可。

### *QueueEndpoint*

- 以队列的资源描述作为Endpoint。
- 格式：`acs:mns:{REGION}:{AccountID}:queues/{QueueName}`。
- 限制：只支持推送到主题同地域同账号的队列中。
- MNS直接将消息写入到指定的队列中，用户通过receive message接口读取消息，具体使用方式请参见[队列推送](#)。

## 1.8. NotifyStrategy

NotifyStrategy描述了消息服务MNS向Endpoint推送消息出现错误时的重试策略。

策略	描述	参数
退避重试	重试3次，每次重试的间隔时间是10秒到20秒之间的随机值。	BACKOFF_RETRY
指数衰减重试	重试176次，每次重试的间隔时间指数递增至512秒，总计重试时间为1天。每次重试的具体间隔为：1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 512...512（共167个512）。	EXPONENTIAL_DECAY_RETRY

## 1.9. NotifyContentFormat

NotifyContentFormat用于指定MNS向Endpoint推送消息时，消息内容的格式。

### XML

- 消息体为XML格式，包含消息正文和消息属性。
- HttpEndpoint和QueueEndpoint支持该格式。
- 消息示例：

```
<?xml version="1.0" encoding="utf-8"?>
  <Notification xmlns="http://mns.aliyuncs.com/doc/v1/">
    <TopicOwner>TopicOwner</TopicOwner>
    <TopicName>TopicName</TopicName>
    <Subscriber>Subscriber</Subscriber>
    <SubscriptionName>SubscriptionName</SubscriptionName>
    <MessageId>6CC4D900CA59A2CD-1-15180534A8F-200000002</MessageId>
    <Message>{1:"a", 2:"b"}</Message>
    <MessageMD5>F1E92841751D795AB325861034B5CB55</MessageMD5>
    <MessageTag>important</MessageTag>
    <PublishTime>1449556920975</MessagePublishTime>
  </Notification>
```

### JSON

- 消息体为JSON格式，包含消息正文和消息属性。
- HttpEndpoint和QueueEndpoint支持该格式。
- 消息示例：

```
{
  "TopicOwner":"TopicOwner",
  "TopicName":"TopicName",
  "Subscriber":"Subscriber",
  "SubscriptionName":"SubscriptionName",
  "MessageId":"6CC4D900CA59A2CD-1-15180534A8F-200000002",
  "Message":"xxxxx",
  "MessageMD5":"F1E92841751D795AB325861034B5CB55",
  "MessageTag":"important",
  "PublishTime":"1449556920975"
}
```

### ***SIMPLIFIED***

- 消息体即用户发布的消息，不包含任何属性信息。
- HttpEndpoint、QueueEndpoint和MailEndpoint均支持该格式。
- 消息体示例：

```
{1:"a", 2:"b"}
```

## 2. 调用方式

### 2.1. 请求结构

本文介绍了API调用中请求的服务地址、通信协议、请求方法、请求组成和请求编码。

MNS服务在多个地域部署，每个地域分别提供了公网访问地址和阿里云内网（ECS经典网络或VPC）访问地址，如下表所示。

#### 服务地址

地域	公网	阿里云内网（ECS经典网络或VPC）
华北1	http(s)://{AccountId}.mns.cn-qingdao.aliyuncs.com	http://{AccountId}.mns.cn-qingdao-internal.aliyuncs.com
华北2	http(s)://{AccountId}.mns.cn-beijing.aliyuncs.com	http://{AccountId}.mns.cn-beijing-internal.aliyuncs.com
华东1	http(s)://{AccountId}.mns.cn-hangzhou.aliyuncs.com	http://{AccountId}.mns.cn-hangzhou-internal.aliyuncs.com
华东2	http(s)://{AccountId}.mns.cn-shanghai.aliyuncs.com	http://{AccountId}.mns.cn-shanghai-internal.aliyuncs.com
华南1	http(s)://{AccountId}.mns.cn-shenzhen.aliyuncs.com	http://{AccountId}.mns.cn-shenzhen-internal.aliyuncs.com
亚太（新加坡）	http(s)://{AccountId}.mns.ap-southeast-1.aliyuncs.com	http://{AccountId}.mns.ap-southeast-1-internal.aliyuncs.com
美国硅谷	http(s)://{AccountId}.mns.us-west-1.aliyuncs.com	http://{AccountId}.mns.us-west-1-internal.aliyuncs.com
亚太东北1（日本）	http(s)://{AccountId}.mns.ap-northeast-1.aliyuncs.com	http://{AccountId}.mns.ap-northeast-1-internal.aliyuncs.com
亚太东南2（悉尼）	http(s)://{AccountId}.mns.ap-southeast-2.aliyuncs.com	http://{AccountId}.mns.ap-southeast-2-internal.aliyuncs.com
欧洲中部1（法兰克福）	http(s)://{AccountId}.mns.eu-central-1.aliyuncs.com	http://{AccountId}.mns.eu-central-1-internal.aliyuncs.com
中东东部1（迪拜）	http(s)://{AccountId}.mns.me-east-1.aliyuncs.com	http://{AccountId}.mns.me-east-1-internal.aliyuncs.com
华东1金融云	暂无	http://{AccountId}.mns.cn-hzjbp-a-internal.aliyuncs.com



地域	公网	阿里云内网（ECS经典网络或VPC）
华东2金融云	暂无	http://{AccountId}.mns.cn-shanghai-finance-internal.aliyuncs.com
华南1金融云	暂无	http://{AccountId}.mns.cn-shenzhen-finance-internal.aliyuncs.com

AccountID是阿里云注册用户的账号ID，在阿里云官网可以查看。为了帮助保护用户的数据安全，MNS在公网提供了HTTPS的接口，只需要把公网访问地址的http://换成https://即可。

## 通信协议

支持通过HTTP协议进行请求通信。

## 请求方法

使用HTTP的PUT、POST、GET、DELETE等HTTP Method发送不同的请求，详情请参见[队列接口规范](#)和[主题接口规范](#)。

## 请求组成

向MNS发送HTTP请求完成某个API接口操作，发送的请求需要带上正确的请求参数、请求头以及请求正文。

## 请求编码

请求及返回结果都使用UTF-8字符集进行编码。

# 2.2. 公共参数

本文介绍在API调用过程中涉及到的公共请求参数和公共返回参数。

## 公共请求参数

参数名称	说明	是否必选
Authorization	验证字符串，由MNS + 空格 + AccessKeyId + : + signature构成，详见本文在Head中包含签名部分	是
Content-Length	HTTP消息体的长度	是
Content-Type	请求内容的MIME类型，目前请求仅支持 text或xml 格式	是
Content-MD5	HTTP消息体的MD5值，具体格式请参见 <a href="https://tools.ietf.org/html/rfc1864">https://tools.ietf.org/html/rfc1864</a>	否

参数名称	说明	是否必选
Date	请求的构造时间，目前只支持GMT格式，如果和MNS的服务器时间前后差异超过15分钟将返回本次请求非法	是
Host	从阿里云官网获取AccountId，从API文档中获取各地域MNS访问地址，格式如下： 下：\$AccountId.mns.cn-hangzhou.aliyuncs.com	针对HTTP/1.1为必选，针对HTTP/1.0为可选
x-mns-version	调用MNS接口的版本号，当前版本为2015-06-06	是
x-mns-date	Date替代字段，用于解决部分浏览器上，用户程序无法设置HTTP请求Date字段的场景	否

## 公共返回参数

参数名称	说明
Content-Length	HTTP消息体返回的长度
Connection	HTTP连接状态
Date	响应的返回时间，GMT时间格式
Server	请求响应的MNS服务器名
x-mns-request-id	此次Request操作的编号
x-mns-version	MNS接口的版本编号，当前版本是2015-06-06

## 2.3. 返回结果

您可以根据返回数据判断API服务是否调用成功。

调用API服务后返回数据采用统一格式。返回的HTTP状态码为2xx，说明调用成功；返回的HTTP状态码为4xx或5xx，说明调用失败。调用成功返回的数据格式为XML。本文中的返回示例为了便于用户查看，做了格式化处理，实际返回结果没有进行换行、缩进等处理。

XML返回结果包括请求是否成功信息和具体的业务数据。示例如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<根节点 xmlns="http://mns.aliyuncs.com/doc/v1/">
<!--返回的子结点-->
</根节点>
```

## 错误结果

调用接口出错后，将不会返回结果数据，具体的错误信息请参见[错误码](#)。

当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求ID: RequestId和一个您本次请求访问的站点ID: HostId。调用方找不到错误原因，可以联系阿里云客服，提供HostId和RequestId以便客服尽快帮您解决问题。

## 2.4. 请求签名机制

本文介绍API请求的签名机制。

### 签名机制

AccessKeyId和AccessKeySecret由阿里云官方颁发给访问者（可以通过[阿里云AK管理控制台](#)申请和管理），其中AccessKeyId用于标识访问者的身份，AccessKeySecret是用于加密签名字符串和服务器端验证签名字符串的密钥，这两个参数必须严格保密。

消息服务MNS服务会对每个访问的请求进行验证，每个请求向消息服务MNS提交时，都需要在该请求的Header中包含签名（Authorization）。消息服务MNS通过使用AccessKeyId和AccessKeySecret进行对称加密的方法来验证请求的发送者身份。如果计算结果和提供的验证码一致，那么该请求有效；否则，消息服务MNS将拒绝处理这次请求，并返回HTTP状态码403。

您必须在HTTP请求中增加Authorization（授权）的Head来包含签名信息，表明这个消息已被授权。格式为：`Authorization: MNS AccessKeyId:Signature`。

Authorization计算方法如下：

您可以使用[签名验证工具](#)来验证您生成的签名。

```
Authorization = base64(hmac-sha1(HTTP_METHOD + "\n"
    + CONTENT-MD5 + "\n"
    + CONTENT-TYPE + "\n"
    + DATE + "\n"
    + CanonicalizedMNSHeaders
    + CanonicalizedResource))
```

- `HTTP_METHOD`：大写的HTTP Method。

例如：PUT、GET、POST、DELETE。

- `Content-Md5`：请求内容数据的MD5值。

如果请求的Header中没有传Content-MD5，则此处填入空串。

- `CONTENT-TYPE`：请求内容的类型。
- `DATE`：此次操作的时间。

此参数不能为空，目前只支持GMT格式。如果请求时间和消息服务MNS服务器时间相差超过15分钟，消息服务MNS会判定此请求不合法，返回错误码400，错误信息及错误码详见[错误响应格式](#)。格式为：`Thu,07 Mar 2012 18:49:58 GMT`。如果用 `x-mns-date` 替代DATE，则DATE不能填空，需用 `x-mns-date` 的值替换。

- CanonicalizedMNSHeaders: HTTP中的 `x-mns-` 开始的字段组合。

**说明** CanonicalizedMNSHeaders (即 `x-mns-` 开头的head) 在签名验证前需要符合以下规范:

- head的名字需要变成小写。
- head自小到大排序。
- 分割head name和value的冒号前后不能有空格。
- 每个Head之后都有一个 `\n` , 如果没有以 `x-mns-` 开头的head, 则在签名时 CanonicalizedMNSHeaders就设置为空。

- CanonicalizedResource表示HTTP所请求资源的URI。示例值: `/queues/$queueName?metaOverride=true`。

**说明**

- 用来签名的字符串为UTF-8格式。
- 签名的方法用RFC 2104中定义的HMAC-SHA1方法, 其中Key为AccessKeySecret。
- content-type和content-md5在请求中不是必须的, 参数没有指定的情况下, 请使用 `"` 代替。

## 示例

- 请求示例:

```
PUT /queues/$queueName?metaOverride=true HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
Authorization: MNS 15B4D3461F177624****:xQE0diMbL****f3YB+FIEXAMPLE=

<?xml version="1.0" encoding="UTF-8" ?>
<Queue xmlns="http://mns.aliyuncs.com/doc/v1/">
  <VisibilityTimeout >60</VisibilityTimeout>
  <MaximumMessageSize>1024</MaximumMessageSize>
  <MessageRetentionPeriod>120</MessageRetentionPeriod>
  <DelaySeconds>30</DelaySeconds>
</Queue>
```

- 返回示例:

- 如果传入的AccessKeyId不存在或disabled，返回403 Forbidden。

返回示例一：

```
Content-Type: text/xml
Content-Length: 314
Date: Wed, 18Mar 2012 08:04:06 GMT
x-mns-request-id: 512B2A634403E52B1956****

<?xml version="1.0" encoding="utf-8"?>
<Error xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Code>AccessIDAuthError</Code>
  <Message>
    AccessID authentication fail, please check your AccessID and retry.
  </Message>
  <RequestId>512B2A634403E52B1956****</RequestId>
  <HostId>mns.cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

- 如果签名验证的时候，Header中没有传入Date或者格式不正确，返回403 Forbidden。

返回示例二：

```
Content-Type: text/xml
Content-Length: 274
Date: Wed, 18Mar 2012 08:04:06 GMT
x-mns-request-id: 512B2A634403E52B1956****

<?xml version="1.0" encoding="UTF-8"?>
<Error xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Code>InvalidArgument</Code>
  <Message>Date Header is invalid or missing.</Message>
  <RequestId>7E1A5CF258F535884403****</RequestId>
  <HostId>mns.cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

- 传入请求的时间在消息服务MNS服务器当前时间之后的15分钟以内，否则返回408超时。

返回示例三：

```
Content-Type: text/xml
Content-Length: 283
Date: Wed, 11 May 2011 09:01:51 GMT
x-mns-request-id: 512B2A634403E52B1956****

<?xml version="1.0" encoding="UTF-8"?>
<Error xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Code>TimeExpired</Code>
  <Message>
    The http request you sent is expired.
  </Message>
  <RequestId>512B2A634403E52B1956****</RequestId>
  <HostId>mns.cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

## 2.5. Endpoint签名认证

您可以通过推送请求头中的x-mns-signing-cert-url获取签名证书，并根据相应的方法来验证该请求是否由消息服务MNS系统发出，防止恶意请求造成负面影响。

在消息服务MNS推送请求头中，Authorization字段的值是消息服务MNS根据待签名字符串，用SHA1-RSA签名算法生成签名。Endpoint可以使用公钥对签名进行验证，具体的验证方法如下：

### 步骤一：获取X509证书

在消息服务MNS发送给Endpoint的HTTP请求头中，x-mns-signing-cert-url指定了签名证书的地址，您需要通过Base64解码，获取该签名文件URL地址，再从中提取出签名的公钥。

### 步骤二：计算待签名字符串

```
VERB + "\n"
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ DATE + "\n"
+ CanonicalizedMNSHeaders
+ CanonicalizedResource
```

- VERB：HTTP的方法。
- CONTENT-MD5：请求内容数据的MD5值。
- CONTENT-TYPE：请求内容的类型，对应的值为全小写。
- DATE：此次操作的时间，不能为空，目前只支持GMT格式。

- CanonicalizedMNSHeaders: HTTP请求头中的x-mns-开头的字段组合。
- CanonicalizedResource: HTTP请求的相对地址, 不能为空。

待签名字符串示例:

```
POST
ZDgxNjY5ZjFIMDQ5MGM0YWMwMWE5ODlmZDVIYmQxYjI=
text/xml; charset=utf-8
Wed, 25 May 2016 10:46:14 GMT
x-mns-request-id:57458276F0E3D56D7C00****
x-mns-signing-cert-url:aHR0cDovL21uc3Rlc3Qub3NzLWNuLWVhbmd6aG91LmFsaXl1bmNzLmNvbS94NTA5X
3B1YmxpY19jZXJ0aWZpY2F0ZS5wZW0=
x-mns-version:2015-06-06
/notifications
```

### 步骤三: Authorization解密

对Authorization签名字段进行Base64解码后, 使用从步骤一中提取的公钥对其进行解密。

### 步骤四: 认证

比较步骤二生成的待签名字符串与步骤三解密的结果是否一致。如果一致, 说明请求来自消息服务MNS, 访问合法。

#### 注意

CanonicalizedMNSHeaders在签名验证前需要符合以下规范:

- Head的名字需要变成小写。
- Head从小到大排序。
- 分割Headname和value的冒号前后不能有空格。
- 每个Head之后都有一个\n, 如果没有以x-mns-开头的Head, 则在签名时CanonicalizedMNSHeaders就设置为空。

其他认证说明:

- 用来签名的字符串为UTF-8格式。
- 签名的方法用RFC 3447中定义的sha1WithRSAEncryption方法。
- Base64是指使用Base64算法转码文本。

### Java示例代码

```
public class SignDemo {
    private Boolean authenticate(String method, String uri, Map<String, String> headers) {
        try {
            //获取证书的URL。
            if (!headers.containsKey("x-mns-signing-cert-url")) {
                System.out.println("x-mns-signing-cert-url Header not found");
            }
        }
    }
}
```

```
        return false;
    }
    String cert = headers.get("x-mns-signing-cert-url");
    if (cert.isEmpty()) {
        System.out.println("x-mns-signing-cert-url empty");
        return false;
    }
    cert = new String(Base64.decodeBase64(cert));
    System.out.println("x-mns-signing-cert-url:\t" + cert);

    //根据URL获取证书，并从证书中获取公钥。
    URL url = new URL(cert);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    DataInputStream in = new DataInputStream(conn.getInputStream());
    CertificateFactory cf = CertificateFactory.getInstance("X.509");
    Certificate c = cf.generateCertificate(in);
    PublicKey pk = c.getPublicKey();

    //获取待签名字符串。
    String str2sign = getSignStr(method, uri, headers);
    System.out.println("String2Sign:\t" + str2sign);

    //对Authorization字段做Base64解码。
    String signature = headers.get("Authorization");
    byte[] decodedSign = Base64.decodeBase64(signature);

    //认证。
    java.security.Signature signetcheck = java.security.Signature.getInstance("SHA1withRSA");
    signetcheck.initVerify(pk);
    signetcheck.update(str2sign.getBytes());
    Boolean res = signetcheck.verify(decodedSign);
    return res;
} catch (Exception e) {
    e.printStackTrace();
    return false;
}
}

private String getSignStr(String method, String uri, Map<String, String> headers) {
    StringBuilder sb = new StringBuilder();
    sb.append(method);
```



```
sb.append("\n");
sb.append(safeGetHeader(headers, "Content-md5"));
sb.append("\n");
sb.append(safeGetHeader(headers, "Content-Type"));
sb.append("\n");
sb.append(safeGetHeader(headers, "Date"));
sb.append("\n");

List<String> tmp = new ArrayList<String>();
for (Map.Entry<String, String> entry : headers.entrySet()) {
    if (entry.getKey().startsWith("x-mns-")) {
        tmp.add(entry.getKey() + ":" + entry.getValue());
    }
}
Collections.sort(tmp);

for (String kv : tmp) {
    sb.append(kv);
    sb.append("\n");
}

sb.append(uri);
return sb.toString();
}

private String safeGetHeader(Map<String, String> headers, String name) {
    if (headers.containsKey(name)) {
        return headers.get(name);
    } else {
        return "";
    }
}

public static void main(String[] args) {
    SignDemo sd = new SignDemo();
    Map<String, String> headers = new HashMap<String, String>();
    headers.put("Authorization", "Mko2Azg9fhCw8qR6G7AeAFMyzjO9qn7LDA5/t9E+6X5XURXTqBUuHPK+K5
5UNhrnIE2UdDkRrwDxsaDP5ajQdg==");
    headers.put("Content-md5", "M2ViOTE2ZDEyOTlkODBjMjVhNzYjNhNWl3ZWQ1M2E=");
    headers.put("Content-Type", "text/xml;charset=utf-8");
```

```
headers.put("Date", "Tue, 23 Feb 2016 09:41:06 GMT");
headers.put("x-mns-request-id", "56CC2932F0E3D5BD530685CB");
headers.put("x-mns-signing-cert-url", "aHR0cDovL21uc3Rlc3Qub3NzLWNuLWVhbmR6aG91LmFsaXl1bm
NzLmNvbS94NTA5X3B1YmtpY19jZlZlZ0aWZpY2F0ZS5wZW0=");
headers.put("x-mns-version", "2015-06-06");
Boolean res = sd.authenticate("POST", "/notifications", headers);
System.out.println("Authenticate result:" + res);
}
}
```

## 3. 队列接口规范

### 3.1. RESTful API概览

本文列举了消息服务MNS提供的API。

MNS API

#### 队列管理

API	说明	支持方式
CreateQueue	创建队列	控制台、API调用
SetQueueAttributes	修改队列属性	
GetQueueAttributes	获取队列属性	
DeleteQueue	删除队列	
ListQueue	获取队列列表	

#### 消息管理

API	说明	支持方式
SendMessage	发送消息	控制台、API调用
BatchSendMessage	批量发送消息	API调用
ReceiveMessage	消费消息	控制台、API调用
BatchReceiveMessage	批量消费消息	API调用
DeleteMessage	删除消息	控制台、API调用
BatchDeleteMessage	批量删除消息	API调用
PeekMessage	查看消息	API调用
BatchPeekMessage	批量查看消息	API调用
ChangeMessageVisibility	修改消息下次可消费时间	API调用

#### 更多信息

以上列举的部分可通过控制台完成的操作，详情请参见[队列控制台操作帮助](#)。

### 3.2. 队列管理

## 3.2.1. CreateQueue

使用CreateQueue接口来创建一个新的队列。

Name	Value
API	CreateQueue
RAM授权操作	mns:CreateQueue
资源	acs:mns:\$region:\$accountid:/queues/\$queueName

### 描述

队列名称是一个不超过255个字符的字符串，必须以字母或数字为首字符，剩余部分可以包含字母、数字和连字符（-）。

队列实例默认上限为1000个，如有特殊需求可提交工单提高队列实例上限值。

### Request

Request由以下几个部分组成：

- 请求行

```
PUT /queues/$queueName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML中包含创建队列的属性，这些属性都为可选属性。

参数名称	说明	参数值
DelaySeconds	发送到该Queue的所有消息默认将以 <code>DelaySeconds</code> 参数指定的秒数延后可被消费，单位为秒。	0s~604800s（7天）的某个整数值，默认值为0s。
MaximumMessageSize	发送到该Queue的消息体的最大长度，单位为Byte。	1024（1 KB）~65536（64 KB）的某个整数值，默认值为65536（64 KB）。
MessageRetentionPeriod	消息在该Queue中最长的存活时间，从发送到该队列开始经过此参数指定的时间后，不论消息是否被取出过都将被删除，单位为秒。	60s（1分钟）~604800s（7天）二点某个整数值，默认值345600s（4天）。
VisibilityTimeout	消息从该Queue中取出后从Active状态变成Inactive状态后的持续时间，单位为秒。	1s~43200s（12小时）的某个值整数值，默认为30s。

参数名称	说明	参数值
PollingWaitSeconds	当Queue中没有消息时，针对该Queue的ReceiveMessage请求最长的等待时间，单位为秒。	0s~30s的某个整数值，默认为0s。
LoggingEnabled	是否开启日志管理功能。	<ul style="list-style-type: none"> <li>True: 启用。</li> <li>False: 停用。</li> </ul> 默认为False。

**说明**

- 只要Queue设置了 `PollingWaitSeconds`，针对该Queue的所有（Batch）ReceiveMessage请求在队列无消息时，都将默认进入到Polling等待状态（即长轮询）。如果在PollingWaitSeconds时间内一直保持无消息，则返回MessageNotExist；如果在PollingWaitSeconds时间内有新的消息进入到Queue中，则唤醒相应的ReceiveMessage请求，返回消息。
- 进入长轮询后，建议您降低长轮询的并发数。目前MNS服务端会根据用户的长轮询数量、长轮询等待时间、访问IP数量等因素动态设置长轮询防攻击并发上限值。当队列无消息时，超过长轮询上限值的请求将无法进入到监听状态，并直接返回404MessageNotExist（按请求量正常计费）。如果您有临时提升长轮询上限值的需求，请及时提交工单，避免产生不必要的费用支出。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 201 Created

在同一个所有者名下的队列不能重名，当出现创建重名队列时，如果创建队列的属性与已经存在的队列属性相同，返回204（No Content）；否则返回409（Conflict）。

- 特有Response Header

参数名称	说明
Location	返回创建的QueueURL，格式如下： <code>http://\$AccountId.mns.&lt;Region&gt;.aliyuncs.com/queues/\$queueName</code> 。

- Response Body

无。

## 示例

请求示例：

```

PUT /queues/$queueName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
x-mns-version: 2015-06-06
Authorization:MNS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=

<?xml version="1.0" encoding="UTF-8"?>
<Queue xmlns="http://mns.aliyuncs.com/doc/v1/">
  <VisibilityTimeout>60</VisibilityTimeout>
  <MaximumMessageSize>65536</MaximumMessageSize>
  <MessageRetentionPeriod>1209600</MessageRetentionPeriod>
  <LoggingEnabled>True</LoggingEnabled>
</Queue>

```

返回示例：

```

HTTP/1.1 201 Created
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06
Location: http://$AccountId.mns.cn-hangzhou.aliyuncs.com/queues/$queueName

```

## Special error

错误码	错误信息	状态码
QueueAlreadyExist	The queue you want to create is already exist.	409
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400
QueueNumExceededLimit	The number of the queues you created has exceeded the limit.	400

## 3.2.2. SetQueueAttributes

使用Set QueueAttributes接口来修改队列属性。

Name	Value
API	SetQueueAttributes
RAM授权操作	mns:SetQueueAttributes

Name	Value
资源	acs:mns:\$region:\$accountid:/queues/\$queueName

## Request

Request由以下几个部分组成：

- 请求行

```
PUT /queues/$queueName?metaoverride=true HTTP/1.1
```

- URI参数

`metaoverride=true` 将会修改消息队列的属性。

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML中包含创建Queue的属性，这些属性都为可选属性。

参数名称	说明	参数值
DelaySeconds	发送到该Queue的所有消息默认将以 <code>DelaySeconds</code> 参数指定的秒数延后可被消费，单位为秒。	0s~604800s（7天）的某个整数值。
MaximumMessageSize	发送到该Queue的消息体的最大长度，单位为Byte。	1024（1KB）~65536（64KB）的某个整数值。
MessageRetentionPeriod	消息在该Queue中最长的存活时间，从发送到该队列开始经过此参数指定的时间后，不论消息是否被取出过都将被删除，单位为秒。	60s（1分钟）~604800s（7天）某个整数值。
VisibilityTimeout	消息从该Queue中取出后从Active状态变成Inactive状态后的持续时间，单位为秒。	1s~43200s（12小时）的某个值整数值。
PollingWaitSeconds	当Queue消息量为空时，针对该Queue的ReceiveMessage请求最长的等待时间，单位为秒。	0s~30s的某个整数值。
LoggingEnabled	是否开启日志管理功能。	<ul style="list-style-type: none"> <li>o True: 启用。</li> <li>o False: 停用。</li> </ul> 默认为False。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

**HTTP/1.1 204 No Content**

- 特有Response Header  
无，请参见公共参数。
- Response Body  
无。

**示例**

请求示例：

```
PUT /queues/$queueName?Metaoverride=true HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE=

<?xml version="1.0" encoding="UTF-8"?>
<Queue xmlns="http://mns.aliyuncs.com/doc/v1/">
  <VisibilityTimeout>60</VisibilityTimeout>
  <MaximumMessageSize>1024</MaximumMessageSize>
  <MessageRetentionPeriod>120</MessageRetentionPeriod>
  <DelaySeconds>30</DelaySeconds>
  <LoggingEnabled>True</LoggingEnabled>
</Queue>
```

返回示例：

```
HTTP/1.1 204 No Content
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06
```

**Special Error**

错误码	错误信息	状态码
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400
QueueNotExist	The queue name you provided is not exist.	404

**3.2.3. GetQueueAttributes**



使用GetQueueAttributes接口来获取某个已创建队列的属性。

Name	Value
API	GetQueueAttributes
RAM授权操作	mns:GetQueueAttributes
资源	acs:mns:\$region:\$accountid:/queues/\$queueName

## 描述

该接口用于获取某个已创建队列的属性，返回属性除了创建队列时设置的可设置属性外，还可以取到队列创建时间、队列属性修改最后时间以及队列中的各类消息统计数（近似值）。

## Request

Request由以下几个部分组成：

- 请求行

```
GET /queues/$queueName HTTP/1.1
```

- 特有Request Header

无，请参见公共参数。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 200 OK
```

- 特有Response Header

无，请参见公共参数。

- Response Body

返回的结果为XML格式，格式为：`<Queue>Attribute Element List</Queue>`

属性元素列表包括：

参数名称	说明
QueueName	Queue的名称。
CreateTime	Queue的创建时间，从1970-1-1 00:00:00到现在的秒值。
LastModifyTime	修改Queue属性信息最近时间，从1970-1-1 00:00:00到现在的秒值。

参数名称	说明
DelaySeconds	发送消息到该Queue的所有消息默认将以 <code>DelaySeconds</code> 参数指定的秒数延后可被消费，单位为秒。
MaximumMessageSize	发送到该Queue的消息体的最大长度，单位为Byte。
MessageRetentionPeriod	消息在该Queue中最长的存活时间，从发送到该队列开始经过此参数指定的时间后，不论消息是否被取出过都将被删除，单位为秒。
PollingWaitSeconds	当Queue消息量为空时，针对该Queue的 <code>ReceiveMessage</code> 请求最长的等待时间，单位为秒。
ActiveMessages	在该Queue中处于Active状态的消息总数，为近似值。
InactiveMessages	在该Queue中处于Inactive状态的消息总数，为近似值。
DelayMessages	在该Queue中处于Delayed状态的消息总数，为近似值。
LoggingEnabled	是否开启日志管理功能。 <ul style="list-style-type: none"> <li>◦ True: 启用。</li> <li>◦ False: 停用。</li> </ul>

## 示例

请求示例：

```
GET /queues/$queueName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE=
```

返回示例：

```

HTTP/1.1 200 OK
Content-Type:text/xml;charset=utf-8
Content-Length:145
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Queue xmlns="http://mns.aliyuncs.com/doc/v1/">
  <QueueName>$queueName </QueueName ><!--以实际Queue Name返回 -->
  <CreateTime>1250700999</CreateTime >
  <LastModifyTime>1250700999</LastModifyTime>
  <VisibilityTimeout>60</VisibilityTimeout >
  <MaximumMessageSize>65536</MaximumMessageSize>
  <MessageRetentionPeriod>65536</MessageRetentionPeriod>
  <DelaySeconds>30</DelaySecond>
  <PollingWaitSeconds>0</PollingWaitSeconds>
  <InactiveMessages>0</InactiveMessages>
  <ActiveMessages>20</ActiveMessages>
  <DelayMessages>0</DelayMessages>
  <LoggingEnabled>True</LoggingEnabled>
</Queue>
    
```


### Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404

## 3.2.4. DeleteQueue

使用DeleteQueue接口删除已创建的队列。

Name	Value
API	DeleteQueue
RAM授权操作	mns:DeleteQueue
资源	acs:mns:\$region:\$accountid:/queues/\$queueName

 **注意** 请谨慎使用此接口，一旦删除队列成功，队列内所有消息也一并删除，且不可恢复。

## Request

Request由以下几个部分组成：

- 请求行

```
DELETE /queues/$queueName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
DELETE /queues/$queueName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE=
```

返回示例：

```
HTTP/1.1 204 No Content
x-mns-request-id: 512B2A634403E52B1956133E
x-mns-version: 2015-06-06
```

## Special Error

本操作无特有的错误，主要是权限验证类和参数设置类错误，详情请参见[错误码](#)。


### 3.2.5. ListQueue

使用ListQueue接口来列出AccountId下的队列列表，可分页获取数据。

Name	Value
API	ListQueue
RAM授权操作	mns:ListQueue
资源	acs:mns:\$region:\$accountid:/queues

## 描述

该接口用于列出AccountId下的队列列表，可分页获取数据。返回结果中只包含QueueURL属性，如需进一步获取消息队列的属性可以通过GetQueueAttributes接口获取。如果只是想获取特定前缀的队列列表，在调用此接口时指定 x-mns-prefix 参数，返回对队列名称的前缀匹配结果。

 **说明** 如果本次请求结果返回的消息队列总数超过本次请求的 x-mns-ret-number 参数值，返回的XML结果中包含的 NextMarker 表示下一页开始位置，在下次请求时将该值赋予 x-mns-marker 参数以返回下一页的结果。

## Request

Request由以下几个部分组成：

- 请求行

GET /queues HTTP/1.1

- 特有Request Header

参数名称	是否必选	说明
x-mns-marker	否	请求下一个分页的开始位置，一般从上次分页结果返回的NextMarker获取。
x-mns-ret-number	否	单次请求结果的最大返回个数，可以取1~1000范围内的整数值，默认值为1000。
x-mns-prefix	否	按照该前缀开头的queueName进行查找。

其它参数请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

返回XML形式的结果，顶层Queues元素下包含多个Queue元素，Queue元素中包含QueueURL元素，QueueURL表示Queue的可访问URL地址。如果返回结果超过 `x-mns-ret-number` 的参数值，包含 `NextMarker` 元素表示下个分页的起始位置。

## 示例

请求示例：

```
GET /queues HTTP/1.1
x-mns-marker: xxxx
x-mns-ret-number:2
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 08 Mar 2012 12:00:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE=
```

返回示例：

```
HTTP/1.1 200 OK
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06
<?xml version="1.0" encoding="UTF-8"?>
<Queues xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Queue>
    <QueueURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/queues/queueName1</QueueURL>
  </Queue>
  <Queue>
    <QueueURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/queues/queueName2</QueueURL>
  </Queue>
  <NextMarker> Base64 Encoded Result </NextMarker>
</Queues>
```

## Special error

本操作无特有的错误，主要是权限验证类和参数设置类错误，详情请参见[错误码](#)。

## 3.3. 消息管理


### 3.3.1. SendMessage

使用SendMessage接口来发送消息到指定的队列。

Name	Value
API	SendMessage
RAM授权操作	mns:SendMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

## 描述

该接口用于发送消息到指定的队列，普通消息发送到队列立即被消费者消费。如果生产者发送一个消息不想立即被消费者消费（典型的使用场景为定期任务），生产者可以在发送消息时设置 DelaySeconds 参数。发送带 DelaySeconds 参数值大于0的消息初始状态为Delayed，此时消息不能被消费者消费，只有等 DelaySeconds 时间后消息变成Active状态后才可消费。

 **说明** 发送消息时指定 DelaySeconds 参数的优先级高于延时队列的 DelaySeconds 属性，即两个 DelaySeconds 属性设置的值不同的时候以发送消息时指定的值为准。

## Request

请求消息由请求行、HTTP头和消息体三部分组成：

- 请求行

POST /queues/\$queueName/messages HTTP/1.1

- 特有Request Header

无，请参见公共参数。

- Request Body

Request Body为XML格式，其中包含创建Message的属性。

参数名称	是否必选	说明	参数值
MessageBody	是	消息正文	UTF-8字符集。
DelaySeconds	否	DelaySeconds指定的秒数延后可被消费，单位为秒	0s~604800s（7天）范围内某个整数值，默认值为0s。
Priority	否	指定消息的优先级权值，优先级越高的消息，越容易更早被消费	取值范围1~16（其中1为最高优先级），默认优先级为8。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

**HTTP/1.1 201 Created**

- 特有Response Header

无，请参见公共参数。

- Response Body

Response Body为XML格式，返回Message的MessageId和MessageBodyMD5子元素。

参数名称	说明
MessageId	消息编号，在一个队列唯一。
MessageBodyMD5	消息正文的MD5值。
ReceiptHandle	发送延迟消息后返回的消息句柄。

## 示例

请求示例：

```
POST /queues/$queueName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 18 Mar 2012 12:00:00 GMT
Content-Length:500
Content-Type:text/xml;charset=utf-8
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE

<?xml version="1.0" encoding="UTF-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageBody>Base64 Encoded Result</MessageBody>
  <DelaySeconds>60</DelaySeconds>
  <Priority>1</Priority>
</Message>
```

返回示例：



```

HTTP/1.1 201 Created
Content-Length:120
Content-Type:text/xml;charset=utf-8
x-mns-request-id:512B2A634403E52B1956****
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
<MessageId>5F290C926D472878-2-14D9529****-200000001</MessageId>
<MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
<ReceiptHandle>1-ODU40TkzNDU5My0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
</Message>

```

## Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MalformedXML	The XML you provided was not well-formed.	400
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400

## 3.3.2. BatchSendMessage

使用BatchSendMessage接口来批量发送消息到指定的队列。

Name	Value
API	BatchSendMessage
RAM授权操作	mns:BatchSendMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

### 描述

该接口用于批量发送消息到指定的队列，一次BatchSendMessage最多发送16条消息（一次Batch请求的消息总大小不得超过64 KB）。普通消息发送到队列立即被消费者消费，如果生产者发送一个消息不想立即被消费者消费（典型的使用场景为定期任务），生产者可以在发送消息时设置 `DelaySeconds` 参数。

发送带 `DelaySeconds` 参数值大于0的消息初始状态为Delayed，此时消息不能被消费者消费，只有等 `DelaySeconds` 时间后消息变成Active状态后才可消费。

**说明** 发送消息时指定 `DelaySeconds` 参数的优先级高于延时消息队列的 `DelaySeconds` 属性，即两个 `DelaySeconds` 属性设置的值不同的时候以发送消息时指定的值为准。

## Request

Request由以下部分组成：

- 请求行

`POST /queues/$queueName/messages HTTP/1.1`

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，其中包含多条创建Message的属性。

参数名称	是否必选	说明	参数值
MessageBody	是	消息正文。	UTF-8字符集。
DelaySeconds	否	<code>DelaySeconds</code> 指定的秒数延后可被消费，单位为秒。	0s~604800s（7天）范围内某个整数值，默认值为0s。
Priority	否	指定消息的优先级权值，优先级越高的消息，越容易更早被消费。	取值范围1~16（数值越小优先级越高），默认优先级为8。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

`HTTP/1.1 201 Created`

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

Response Body为XML格式，返回多条Message的MessageId和MessageBodyMD5子元素。

参数名称	说明
MessageId	消息编号，在一个队列唯一。
MessageBodyMD5	消息正文的MD5值。

## 示例

请求示例：

```
POST /queues/$queueName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 18 Mar 2012 12:00:00 GMT
Content-Length:500
Content-Type:text/xml;charset=utf-8
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE

<?xml version="1.0" encoding="UTF-8"?>
<Messages xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Message>
    <MessageBody>Message1 Base64 Encoded Result</MessageBody>
    <DelaySeconds>60</DelaySeconds>
    <Priority>1</Priority>
  </Message>
  <Message>
    <MessageBody>Message2 Base64 Encoded Result</MessageBody>
    <DelaySeconds>60</DelaySeconds>
    <Priority>1</Priority>
  </Message>
</Messages>
```

返回头示例：

- 正常发送所有消息

```
HTTP/1.1 201 Created
Content-Length:120
Content-Type:text/xml;charset=utf-8
x-mns-request-id:512B2A634403E52B1956****
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Messages xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Message>
    <MessageId>5F290C926D472878-2-14D9529A8FA-20000****</MessageId>
    <MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
  </Message>
  <Message>
    <MessageId>5F290C926D472878-2-14D9529A8FA-20000****</MessageId>
    <MessageBodyMD5>377AC5283D8765C9CEE4E0EA353DFC17</MessageBodyMD5>
  </Message>
</Messages>
```

- 只发送了部分消息

```
HTTP/1.1 500
Content-Length:120
Content-Type:text/xml;charset=utf-8
x-mns-request-id:512B2A634403E52B1956****
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Messages xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Message>
    <MessageId>5F290C926D472878-2-14D9529A8FA-20000****</MessageId>
    <MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
  </Message>
  <Message>
    <MessageId>5F290C926D472878-2-14D9529A8FA-20000****</MessageId>
    <MessageBodyMD5>377AC5283D8765C9CEE4E0EA353DFC17</MessageBodyMD5>
  </Message>
  <Message>
    <ErrorCode>InternalServerError</ErrorCode>
    <ErrorMessage>Internal error.</ErrorMessage>
  </Message>
</Messages>
```

## Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MalformedXML	The XML you provided was not well-formed.	400
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400

### 3.3.3. ReceiveMessage

消费者使用ReceiveMessage接口来消费队列中的消息。

Name	Value
API	ReceiveMessage
RAM授权操作	mns:ReceiveMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

#### 描述

该接口用于消费者消费队列中的消息，ReceiveMessage操作会将取得的消息状态变成Inactive，Inactive状态的时间长度由Queue属性 `VisibilityTimeout` 指定，详情请参见[CreateQueue](#)。

消费者在 `VisibilityTimeout` 时间内消费成功后需要调用DeleteMessage接口删除该消息，否则该消息将会重新变成Active状态，又可被消费者重新消费。

#### Request

Request由以下部分组成：

- 请求行

```
GET /queues/$queueName/messages?waitseconds=10 HTTP/1.1
```

- 特有URI参数

参数	是否必选	说明
waitseconds	否	本次ReceiveMessage请求最长的Polling等待时间，单位为秒。

### 说明

- 如果ReceiveMessage请求附带waitseconds参数，在Queue无消息时，此次ReceiveMessage请求进入到Polling状态（即长轮询），等待的时长为waitseconds，在这段时间中，如果Queue有消息写入会返回给用户；如果未设置waitseconds，则默认使用所属Queue的PollingWaitSeconds属性，详情请参见CreateQueue。
- 进入到长轮询后，建议您降低长轮询的并发数。目前消息服务MNS服务端会根据长轮询数量、长轮询等待时间、访问IP数量等因素动态设置长轮询防攻击并发上限值。当队列无消息时，超过长轮询上限值的请求将无法进入到监听状态，并直接返回404MessageNotExist（按请求量正常计费）。如果您有临时提升长轮询上限值的需求，请及时提工单联系客服，避免产生不必要的费用支出。

- 特有Request Header

无，请参见公共参数。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见公共参数。

- Response Body

返回的结果为XML格式，返回Message消息正文及消息属性。

参数名称	说明
MessageId	消息编号，在一个Queue中唯一。
ReceiptHandle	本次获取消息产生的临时句柄，用于删除和修改处于Inactive消息，NextVisibleTime之前有效。
MessageBody	消息正文。
MessageBodyMD5	消息正文的MD5值。
EnqueueTime	消息发送到队列的时间，从1970年1月1日00:00:00 000开始的毫秒数。
NextVisibleTime	下次可被再次消费的时间，从1970年1月1日00:00:00 000开始的毫秒数。
FirstDequeueTime	第一次被消费的时间，从1970年1月1日00:00:00 000开始的毫秒数。
DequeueCount	总共被消费的次数。

参数名称	说明
Priority	消息的优先级权值。

## 示例

请求示例：

```
GET /queues/$queueName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE
```

返回示例：

```
HTTP/1.1 200 OK
Content-Type=text/xml;charset=utf-8
Content-Length:500
x-mns-request-id:512B2A634403E52B1956****
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageId>5F290C926D472878-2-14D9529A8FA-200000001</MessageId>
  <ReceiptHandle>1-ODU40TkzNDU5My0xNDMyNzI3ODI3LTItOA==</ReceiptHandle>
  <MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
  <MessageBody>This is a test message</MessageBody>
  <EnqueueTime>1250700979248</EnqueueTime>
  <NextVisibleTime>1250700799348</NextVisibleTime>
  <FirstDequeueTime>1250700779318</FirstDequeueTime >
  <DequeueCount>1</DequeueCount >
  <Priority>8</Priority>
</Message>
```

## Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MessageNotExist	Message not exist.	404

### 3.3.4. BatchReceiveMessage

使用BatchReceiveMessage接口来批量消费队列的消息。

Name	Value
API	BatchReceiveMessage
RAM授权操作	mns:BatchReceiveMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

#### 描述

该接口用于消费者批量消费队列的消息，一次BatchReceiveMessage操作最多可以获取16条消息。该操作会将取得的消息状态变成Inactive，Inactive状态的时间长度由Queue的属性 `VisibilityTimeout` 指定。详情请参见 [CreateQueue](#)。

消费者在 `VisibilityTimeout` 时间内消费成功后需要调用DeleteMessage接口删除取得的消息，否则取得的消息将会被重新变成Active状态，又能被消费者重新消费。

#### Request


Request由以下几个部分组成：

- 请求行

```
GET /queues/$queueName/messages?numOfMessages=16&waitseconds=10 HTTP/1.1
```

- 特有URI参数

参数名称	是否必选	说明
numOfMessages	是	本次BatchReceiveMessage最多获取的消息条数。
waitseconds	否	本次ReceiveMessage请求最长的Polling等待时间，单位为秒。

 **说明** 如果BatchReceiveMessage请求附带 `waitseconds` 参数，在Queue无消息时，此次BatchReceiveMessage请求进入到Polling状态（即长轮询），等待时长为 `waitseconds`。如果未设置 `waitseconds`，则默认使用所属Queue的 `PollingWaitSeconds` 属性，详情请参见 [CreateQueue](#)。

- 特有Request Header

无，请参见 [公共参数](#)。

- Request Body

无。

#### Response



返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

返回的结果为XML格式，返回多条Message消息正文及消息属性。

参数名称	说明
MessageId	消息编号，在一个Queue中唯一。
ReceiptHandle	本次获取消息产生的临时句柄，用于删除和修改处于Inactive消息，NextVisibleTime之前有效。
MessageBody	消息正文。
MessageBodyMD5	消息正文的MD5值。
EnqueueTime	消息发送到队列的时间，从1970年1月1日00:00:00 000开始的毫秒数。
NextVisibleTime	下次可被再次消费的时间，从1970年1月1日 00:00:00 000开始的毫秒数。
FirstDequeueTime	第一次被消费的时间，从1970年1月100:00:00 000开始的毫秒数。
DequeueCount	总共被消费的次数。
Priority	消息的优先级权值。

## 示例

请求示例：

```
GET /queues/$queueName/messages?numOfMessages=16 HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE
```

返回示例：

```

HTTP/1.1 200 OK
Content-Type:text/xml;charset=utf-8
Content-Length:500
x-mns-request-id:512B2A634403E52B1956****
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Messages xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Message>
    <MessageId>5F290C926D472878-2-14D9529A8FA-200000001</MessageId>
    <ReceiptHandle>1-ODU40TkzNDU5My0xNDMyNzI3ODI3LTItOA==</ReceiptHandle>
    <MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
    <MessageBody>This is a test message</MessageBody>
    <EnqueueTime>1250700979248</EnqueueTime>
    <NextVisibleTime>1250700799348</NextVisibleTime>
    <FirstDequeueTime>1250700779318</FirstDequeueTime >
    <DequeueCount>1</DequeueCount>
    <Priority>8</Priority>
  </Message>
  <Message>
    <ReceiptHandle>1-ODU40TkzNDU5My0xNDMyNzI3MjQwLTEtOA==</ReceiptHandle>
    <MessageBodyMD5>C5DD56A39F5F7BB8B3337C6D11B6D8C7</MessageBodyMD5>
    <MessageBody>This is a test message</MessageBody>
    <EnqueueTime>1250700979252</EnqueueTime>
    <NextVisibleTime>1250700799350</NextVisibleTime>
    <FirstDequeueTime>1250700779330</FirstDequeueTime >
    <DequeueCount>1</DequeueCount>
    <Priority>8</Priority>
  </Message>
</Messages>

```

## Special Error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MessageNotExist	Message not exist.	404

## 3.3.5. DeleteMessage

使用DeleteMessage接口来删除已经被消费过的消息。

Name	Value
API	DeleteMessage
RAM授权操作	mns:DeleteMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

## 描述

该接口用于删除已经被消费过的消息，消费者需将上次消费后得到的ReceiptHandle作为参数来定位要删除的消息。本操作只有在 NextVisibleTime 之前执行才能成功。如果过了 NextVisibleTime，消息重新变回 Active状态，ReceiptHandle就会失效；删除失败，需重新消费获取新的ReceiptHandle。

## Request

Request由以下几个部分组成：

- 请求行

```
DELETE /queues/$queueName/messages?ReceiptHandle=<receiptHandle> HTTP/1.1
```

- 特有URI参数

参数名称	说明	是否必选
ReceiptHandle	上次消费后返回的消息ReceiptHandle，详情请参见 <a href="#">ReceiveMessage</a> 。	是

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行，HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
DELETE /queues/$queueName/messages?ReceiptHandle=MbZj6wDWli+QEauMZc8ZRv37sIW2iJKq3M9Mx/KS
bkJ0 HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE
```

返回示例：

```
HTTP/1.1 204 No Content
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06
```

## Special Error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400
ReceiptHandleError	The receipt handle you provide is not valid.	400

## 3.3.6. BatchDeleteMessage

使用BatchDeleteMessage接口来批量删除队列多条消息。

Name	Value
API	BatchDeleteMessage
RAM授权操作	mns:BatchDeleteMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

### 描述

该接口用于批量删除队列多条消息，最多可以删除16条消息，每条消息需要传入相应的ReceiptHandle。

建议此接口与BatchReceiveMessage接口搭配使用，效率会更高。批量操作的结果可能同时包含成功和失败子请求。只有全部子请求都成功时，HTTP状态码才是204。当部分失败发生时，需检查响应中每个消息的错误信息。

## Request

Request由以下几个部分组成：

- 请求行

```
DELETE /queues/$queueName/messages HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，其中包含多条消息的ReceiptHandle。

## Response

返回消息由返回状态行，HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 NoContent
```

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
DELETE /queues/$queueName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE

<?xml version="1.0" encoding="UTF-8"?>
<ReceiptHandles xmlns="http://mns.aliyuncs.com/doc/v1/">
  <ReceiptHandle>1-ODU4OTkzNDU5My0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
  <ReceiptHandle>1-ODU4OTkzNDU5NC0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
  <ReceiptHandle>1-ODU4OTkzNDU5NS0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
</ReceiptHandles>
```

返回示例：

- 正常删除所有消息

```
HTTP/1.1 204 No Content
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06
```

- 部分消息删除失败

```
HTTP/1.1 404
Connection:close
Content-Type=text/xml;charset=utf-8
Content-Length:500
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Errors xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Error>
    <ErrorCode>MessageNotExist</ErrorCode>
    <ErrorMessage>Message not exist.</ErrorMessage>
    <ReceiptHandle>1-ODU4OTkzNDU5My0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
  </Error>
  <Error>
    <ErrorCode>MessageNotExist</ErrorCode>
    <ErrorMessage>Message not exist.</ErrorMessage>
    <ReceiptHandle>1-ODU4OTkzNDU5NC0xNDM1MTk3NjAwLTItNg==</ReceiptHandle>
  </Error>
</Errors>
```

## Special Error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400
ReceiptHandleError	The receipt handle you provide is not valid.	400

### 3.3.7. PeekMessage

使用PeekMessage接口来查看消息。

Name	Value
API	PeekMessage
RAM授权操作	mns:PeekMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

## 描述

该接口用于消费者查看消息，PeekMessage与ReceiveMessage不同，PeekMessage不会改变消息的状态。消息被PeekMessage获取后，消息仍然处于Active状态，可被查看或消费。消息被ReceiveMessage获取后消息进入Inactive状态，在 `VisibilityTimeout` 的时间内不可被查看和消费。

## Request

Request由以下几个部分组成：

- 请求行

`GET /queues/$queueName/messages?peekonly=true HTTP/1.1`

- URI参数

`peekonly=true` 表示这次请求只是去查看队列顶部的消息并不会引起消息的状态改变。

- 特有Request Header

无，请参见公共参数。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

`HTTP/1.1 200 OK`

- 特有Response Header

无，请参见公共参数。

- Response Body

返回的结果为XML格式，返回Message消息正文及消息属性。

参数名称	说明
MessageId	消息编号，在一个Queue唯一。
MessageBody	消息正文。
MessageBodyMD5	消息正文的MD5值。

参数名称	说明
EnqueueTime	消息发送到队列的时间，从1970年1月1日00:00:00 000开始的毫秒数。
FirstDequeueTime	第一次被消费的时间，从1970年1月1日00:00:00 000开始的毫秒数，如果 DequeueCount 为0，则与消息的 EnqueueTime 相同。
DequeueCount	总共被消费的次数。
Priority	消息的优先级权值。

## 示例

请求示例：

```
GET /queues/$queueName/messages?peekonly=true HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE
```

返回示例：

```
HTTP/1.1 200 OK
Connection:close
Content-Type=text/xml;charset=utf-8
Content-Length:500
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageId>D6D5F7C9C12D14A4-1-14D953EFC72-200000004</MessageId>
  <MessageBodyMD5>F9360F391579E71CA77BC5D50242FCF4</MessageBodyMD5>
  <MessageBody>This is a test message</MessageBody>
  <EnqueueTime>1250700979248</EnqueueTime>
  <FirstDequeueTime>1250700979348</FirstDequeueTime >
  <DequeueCount>5</DequeueCount>
  <Priority>8<Priority>
</Message>
```

## Special error



错误代码	错误消息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MessageNotExist	Message not exist.	404

### 3.3.8. BatchPeekMessage

使用BatchPeekMessage接口来批量查看消息。

Name	Value
API	BatchPeekMessage
RAM授权操作	mns:BatchPeekMessage
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

#### 描述

该接口用于消费者批量查看消息，一次最多查看16条消息。BatchPeekMessage与BatchReceiveMessage不同，BatchPeekMessage并不会改变消息的状态。消息被BatchPeekMessage获取后，仍然处于Active状态，可被查看或消费；而消息被BatchReceiveMessage获取后，进入Inactive状态，在 `VisibilityTimeout` 的时间内不可被查看和消费。

#### Request

Request由以下几个部分组成：

- 请求行

```
GET /queues/$queueName/messages?peekonly=true&numOfMessages=16 HTTP/1.1
```

- 特有URI参数

参数名称	是否必选	说明
peekonly=true	是	表示这次请求只是去查看队列顶部的消息并不会引起消息的状态改变。
numOfMessages	是	本次BatchPeekMessage最多查看消息条数。

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

无。

#### Response

返回消息由返回状态行，HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

返回的结果为XML格式，返回多条Message消息正文及消息属性。

参数名称	说明
MessageId	消息编号，在一个Queue唯一。
MessageBody	消息正文。
MessageBodyMD5	消息正文的MD5值。
EnqueueTime	消息发送到队列的时间，从1970年1月1日00:00:00 000开始的毫秒数。
FirstDequeueTime	第一次被消费的时间，从1970年1月1日00:00:00 000开始的毫秒数，如果 <code>DequeueCount</code> 为0，则与消息的EnqueueTime相同。
DequeueCount	总共被消费的次数。
Priority	消息的优先级权值。

## 示例

请求示例：

```
GET /queues/$queueName/messages?peekonly=true&numOfMessages=16 HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE
```

返回示例：

```

HTTP/1.1 200 OK
Connection:close
Content-Type=text/xml;charset=utf-8
Content-Length:500
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Messages xmlns="http://mns.aliyuncs.com/doc/v1/">
<Message>
  <MessageId>D6D5F7C9C12D14A4-1-14D953EFC72-200000004</MessageId>
  <MessageBodyMD5>F9360F391579E71CA77BC5D50242FCF4</MessageBodyMD5>
  <MessageBody>This is a test message</MessageBody>
  <EnqueueTime>1250700979248</EnqueueTime>
  <FirstDequeueTime>1250700979348</FirstDequeueTime >
  <DequeueCount>5</DequeueCount>
  <Priority>8<Priority>
</Message>
<Message>
  <MessageId>D6D5F7C9C12D14A4-1-14D953EFC72-200000005</MessageId>
  <MessageBodyMD5>F9360F391579E71CA77BC5D50242FCF4</MessageBodyMD5>
  <MessageBody>This is a test message</MessageBody>
  <EnqueueTime>1250700979250</EnqueueTime>
  <FirstDequeueTime>1250700979352</FirstDequeueTime >
  <DequeueCount>5</DequeueCount>
  <Priority>8<Priority>
</Message>
</Messages>
    
```

### Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
MessageNotExist	Message not exist.	404

### 3.3.9. ChangeMessageVisibility

使用ChangeMessageVisibility接口来修改被消费过并且还处于Inactive状态的消息到下次可被消费的时间。

Name	Value
API	ChangeMessageVisibility
RAM授权操作	mns:ChangeMessageVisibility
资源	acs:mns:\$region:\$accountid:/queues/\$queueName/messages

## 描述

该接口用于修改被消费过并且还处于Inactive状态的消息到下次可被消费的时间，成功修改消息的 VisibilityTimeout 后，返回新的ReceiptHandle。

## Request

请求消息由由以下几个部分组成：

- 请求行

```
PUT /queues/$queueName/messages?receiptHandle=<receiptHandle>&visibilityTimeout=<visibilitytimeout> HTTP/1.1
```

- URI参数

参数名称	是否必选	说明
ReceiptHandle	是	上次消费后返回的消息 ReceiptHandle，详情请参见 <a href="#">ReceiveMessage</a> 。
VisibilityTimeout	是	从现在到下次可被用来消费的时间间隔，单位为秒。

- 特有Request Header

无，请参见 [公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 200 OK
```

- 特有Response Header

无，请参见 [公共参数](#)。

- Response Body

参数名称	说明
ReceiptHandle	本次修改消息 <code>VisibilityTimeout</code> 时间返回的临时句柄，用于删除和修改处于Inactive消息，在 <code>NextVisibleTime</code> 时间之前有效。
NextVisibleTime	下次可被再次消费的时间，从1970年1月1日00:00:0000开始的毫秒数。

## 示例

请求示例：

```
PUT /queues/$queueName/messages
?receiptHandle=MbZj6wDWli+QEauMZc8ZRV37sIW2iJKq3M9Mx/KSbkJ0&visibilityTimeout=50 HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Wed, 28 May 2012 22:32:00 GMT
x-mns-version: 2015-06-06
Authorization: MNS 15B4D3461F177624206A:xQE0diMblRepdf3YB+FIEXAMPLE
```

返回示例：

```
HTTP/1.1 200 OK
x-mns-request-id:512B2A634403E52B1956133E
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<ChangeVisibility xmlns="http://mns.aliyuncs.com/doc/v1/">
  <ReceiptHandle>TbZj6wDWli+9CEauMZc8ZRV37sIW2iJKq3M9Mx/TS1</ReceiptHandle >
  <NextVisibleTime>1250700979298000</NextVisibleTime>
</ChangeVisibility>
```

## Special error

错误码	错误信息	状态码
QueueNotExist	The queue name you provided is not exist.	404
InvalidArgument	The value of Element should between Low and High seconds/bytes.	400
MessageNotExist	Message not exist.	404

## 4.主题接口规范

### 4.1. RESTful API概览

本文列举了消息服务MNS主题模式提供的API。

MNS API

#### 主题管理

API	说明
<a href="#">CreateTopic</a>	创建主题
<a href="#">SetTopicAttributes</a>	修改主题属性
<a href="#">GetTopicAttributes</a>	获取主题属性
<a href="#">DeleteTopic</a>	删除主题
<a href="#">ListTopic</a>	获取主题列表

#### 订阅管理

API	说明
<a href="#">Subscribe</a>	创建订阅
<a href="#">SetSubscriptionAttributes</a>	修改订阅属性
<a href="#">GetSubscriptionAttributes</a>	获取订阅属性
<a href="#">Unsubscribe</a>	删除订阅
<a href="#">ListSubscriptionByTopic</a>	获取主题的订阅列表

#### 消息管理

API	说明
<a href="#">PublishMessage</a>	发布消息

#### 通知管理

API	说明
<a href="#">HttpEndpoint</a>	推送通知

以上列举的API操作还可通过控制台完成。详情请参见[主题控制台操作帮助](#)。

## 4.2. 主题管理

### 4.2.1. CreateTopic

使用CreateTopic接口创建主题。

#### 描述

该接口用于创建一个新的主题。

主题名称是一个不超过255个字符的字符串，必须以字母或数字为首字符，剩余部分可以包含字母、数字和横划线(-)。

#### Request

请求消息由以下几个部分组成：

- 请求行

```
PUT /topics/$TopicName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML中包含创建主题的属性。

参数名称	是否必选	说明	参数值
MaximumMessageSize	否	发送到该主题的消息体最大长度，单位为Byte。	1024(1 KB) ~ 65536(64 KB)范围内的某个整数值，默认值为65536(64 KB)。
LoggingEnabled	否	是否开启日志管理功能。	<ul style="list-style-type: none"> <li>True: 启用。</li> <li>False: 停用。</li> </ul> 默认为False。

#### Response

返回消息由返回状态行，HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 201 Created
```

同一个所有者在同一个地域的主题不能重名，当出现创建重名的主题时，如果创建主题的属性与已经存在的主题属性相同，返回204（No Content），否则返回409（Conflict）。

- 特有Response Header

参数名称	说明

参数名称	说明
Location	返回创建的TopicURL，格式如下： <code>http://\$AccountId.mns.&lt;Region&gt;.aliyuncs.com/topics/\$TopicName</code> 。

其余返回Header请参见[公共参数](#)。

- Response Body  
无。

## 示例

请求示例：

```
PUT /topics/$TopicName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 05:57:57 GMT
x-mns-version: 2015-06-06
Authorization:MNS 15B4D3461F177624206A:eCtaUwfGS0oEfnzJJEhWFxpf2yk=

<?xml version="1.0" encoding="utf-8"?>
<Topic xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MaximumMessageSize>10240</MaximumMessageSize>
  <LoggingEnabled>True</LoggingEnabled>
</Topic>
```

返回示例：

```
HTTP/1.1 201 Created
x-mns-request-id:56667165B2B71C9C16000013
x-mns-version: 2015-06-06
Location: http://$AccountId.mns.cn-hangzhou.aliyuncs.com/topics/$TopicName
```

## Special error

错误代码	错误消息	状态码
TopicAlreadyExist	The topic you want to create is already exists.	409



错误代码	错误消息	状态码
TopicNameLengthError	Topic name length is out of range, should be between 1 and 255.	400

## 4.2.2. SetTopicAttributes

使用SetTopicAttributes接口来修改主题的属性。

### Request

请求消息由以下几个部分组成：

- 请求行

```
PUT /topics/$TopicName?metaoverride=true HTTP/1.1
```

- 特有URI参数

metaoverride=true 将会修改主题的属性。

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML格式中包含设置主题的属性。

参数名称	是否必选	说明	参数值
MaximumMessageSize	否	发送到该Topic的消息体最大长度，单位为Byte。	1024 (1 KB) ~65536 (64 KB) 范围内的某个整数值。
LoggingEnabled	否	是否开启日志管理功能。	<ul style="list-style-type: none"> <li>◦ True: 启用。</li> <li>◦ False: 停用。</li> </ul>

### Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

### 示例

请求示例：

```

PUT /topics/$TopicName?metaoverride=true HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
x-mns-version: 2015-06-06
Date: Tue, 08 Dec 2015 06:01:00 GMT
Authorization:MNS 15B4D3461F177624206A:SeIRcZk3lKiP7bl19WB9Uz07bl8=

<?xml version="1.0" encoding="utf-8"?>
<Topic xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MaximumMessageSize>65536</MaximumMessageSize>
  <LoggingEnabled>True</LoggingEnabled>
</Topic>

```

返回示例：

```

HTTP/1.1 204 No Content
x-mns-request-id:5666721CB2B71C9C1600001B
x-mns-version: 2015-06-06

```

## Special error

错误代码	错误消息	状态码
TopicNotExist	The topic you provided does not exist.	404

## 4.2.3. GetTopicAttributes

使用GetTopicAttributes接口来获取主题的属性。

### 描述

该接口用于获取某个已创建主题的属性，返回属性除创建主题时的可设置属性外，还可以获取主题的消息最长存活时间、主题创建时间等。

### Request

Request的构造主要由以下几个部分组成：

- 请求行

```
GET /topics/$TopicName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

Response Body返回的结果为XML格式，格式为：`<TopicAttributes>Attribute Element List</TopicAttributes>`。

属性元素列表包括以下参数。

参数名称	说明
TopicName	主题名称。
CreateTime	主题的创建时间，从1970-1-1 00:00:00到现在的秒值。
LastModifyTime	修改主题属性信息的最近时间，从1970-1-1 00:00:00到现在的秒值。
MaximumMessageSize	发送到该主题的消息体最大长度，单位为Byte。
MessageRetentionPeriod	消息在主题中最长存活时间，从发送到该主题开始经过此参数指定的时间后，不论消息是否被成功推送给用户都将被删除，单位为秒。
MessageCount	当前该主题中消息数目。
LoggingEnabled	是否开启日志管理功能，取值说明如下： <ul style="list-style-type: none"> <li>◦ True：启用。</li> <li>◦ False：停用。</li> </ul>

## 示例

请求示例：

```
GET /topics/$TopicName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:02:33 GMT
x-mns-version: 2015-06-06
Authorization:MNS 15B4D3461F177624206A:aVZgLZzVchjOtM1aecDQYFuj4Bo=
```

返回示例：

```

HTTP/1.1 200 OK
Content-Type=text/xml;utf-8
Content-Length:145
x-mns-request-id:56667279B2B71C9C1600001D
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="UTF-8"?>
<Topic xmlns=http://mns.aliyuncs.com/doc/v1/" >
  <TopicName>$TopicName</TopicName>
  <CreateTime>1449554277</CreateTime >
  <LastModifyTime>1449554460</LastModifyTime>
  <MaximumMessageSize>65536</MaximumMessageSize>
  <MessageRetentionPeriod>86400</MessageRetentionPeriod>
  <MessageCount>0</MessageCount>
  <LoggingEnabled>True</LoggingEnabled>
</Topic>

```

## Special error


错误代码	错误消息	状态码
TopicNotExist	The topic you provided does not exist.	404

## 4.2.4. DeleteTopic

使用DeleteTopic接口来删除主题。

### 描述

该接口用于删除一个已创建的主题。

 **注意** 使用者必须谨慎使用此接口，一旦删除成功，主题内所有消息一并删除，且不可恢复。所有订阅该主题的Subscription也一并被删除，且不可恢复。

### Request

请求消息由以下几个部分组成：

- 请求行

```
DELETE /topics/$TopicName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

如果主题不存在，返回204，删除操作成功。

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
DELETE /topics/$TopicName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:05:08 GMT
Authorization: MNS 15B4D3461F177624206A:xit63dyKs83PTJF0hx3fd04CWog=
x-mns-version: 2015-06-06
```

返回示例：


```
HTTP/1.1 204 No Content
x-mns-request-id: 56667314B2B71C9C16000021
x-mns-version: 2015-06-06
```

## 4.2.5. ListTopic

使用ListTopic接口列出帐号下的主题列表，可分页获取数据。

### 描述

该接口用于列出帐号下的主题列表，可分页获取数据。如果只是想获取特定的主题列表，在调用接口时指定 `x-mns-prefix` 参数，服务端将返回主题名称与前缀匹配的主题列表。

 **注意** 如果本次请求结果返回的主题总数超过本次请求的 `x-mns-ret-number` 参数值，返回的XML结果中包含的 `NextMarker` 表示下一页开始位置，在下次请求时将该值赋予 `x-mns-marker` 参数以返回下一页的结果。

## Request

Request的构造主要由以下几个部分组成：

- 请求行

GET /topics HTTP/1.1

- 特有Request Header

参数名称	是否必选	说明
x-mns-marker	否	请求下一个分页的开始位置，从上次分页结果返回的 <code>NextMarker</code> 获取。
x-mns-ret-number	否	单次请求结果的最大返回个数，1~1000范围内的整数值，默认值为1000。
x-mns-prefix	否	按照该前缀开头的主题名称进行查找。

其它参数请参见[公共参数](#)。

- Request Body

无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

返回XML形式的结果，顶层 `Topics` 元素下包含多个 `Topic` 元素，`Topic` 元素中包含 `TopicURL` 元素，`TopicURL` 表示Topic的URL访问地址。如果返回结果超过 `x-mns-ret-number` 的参数值，包含 `NextMarker` 元素表示下个分页的起始位置。

## 示例

请求示例：

```
GET /topics HTTP/1.1
x-mns-ret-number:2
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 05:59:22 GMT
Authorization: NS15B4D3461F177624206A:ytChleJt2klpvFxCgVsPNI015k=
x-mns-version: 2015-06-06
```

返回示例：

```

HTTP/1.1 200 OK
x-mns-request-id:566671BAB2B71C9C16000019
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Topics xmlns="http://mns.aliyuncs.com/doc/v1">
  <Topic>
    <TopicURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/topics/$TopicName1</TopicURL>
  </Topic>
  <Topic>
    <TopicURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/topics/$TopicName2</TopicURL>
  </Topic>
  <NextMarker>OTczNjU4MTcvYmRwejd6NXluby8yNjg0Mi9UZjN0VG9waWMTMy8w</NextMarker>
</Topics>

```

## 4.3. 订阅管理

### 4.3.1. Subscribe

使用Subscribe接口来创建Subscription。

#### 描述

该接口用于订阅主题，创建Subscription。Subscription名称是一个不超过255个字符的字符串，必须以字母或者数字为首字符，剩余部分可以包含字母、数字和连字符（-）。

创建Subscription时，需要指定对应的Endpoint，否则不合法。

#### Request

请求消息由以下几部分组成：

- 请求行

```
PUT /topics/$TopicName/subscriptions/$SubscriptionName HTTP/1.1
```

- 特有Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML总包含创建Subscription的属性。

参数名称	是否必选	说明	参数值
------	------	----	-----

参数名称	是否必选	说明	参数值
Endpoint	是	描述此次订阅中接收消息的终端地址。	目前四种Endpoint： <ul style="list-style-type: none"> <li>◦ HttpEndpoint, 必须以http://为前缀。</li> <li>◦ QueueEndpoint, 格式为acs:mns:{REGION}:{AccountID}:queues/{QueueName}。</li> <li>◦ MailEndpoint, 格式为mail:directmail:{MailAddress}。</li> <li>◦ SmsEndpoint, 格式为sms:directsms:anonymous 或 sms:directsms:{Phone}。</li> </ul>
FilterTag	否	描述了该订阅中消息过滤的标签（标签一致的消息才会被推送）。	不超过16个字符的字符串，默认不进行消息过滤。
NotifyStrategy	否	描述了向Endpoint推送消息出现错误时的重试策略。	BACKOFF_RETRY或者EXPONENTIAL_DECAY_RETRY, 默认为BACKOFF_RETRY, 重试策略的具体描述请参见 <a href="#">NotifyStrategy</a> 。
NotifyContentFormat	否	描述了向Endpoint推送的消息格式。	XML、JSON或者SIMPLIFIED, 默认为XML, 消息格式的具体描述请参见 <a href="#">NotifyContentFormat</a> 。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 201

同一个订阅者在同一个主题下的Subscription不能重名，当出现重名时，如果正在创建的Subscription已经存在的Subscription属性完全相同，返回204；否则返回409 (Conflict)。

- 特有Response Header

参数名称	说明



参数名称	说明
Location	返回创建成功的SubscriptionURL，格式如下： http://\$AccountId.mns.cn-hangzhou.aliyuncs.com/topics/\$TopicName/subscriptions/\$SubscriptionName。

其它Response Header请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
PUT /topics/$TopicName/subscriptions/$SubscriptionName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:06:46 GMT
Authorization:MNS 15B4D3461F177624206A:xQE0diMbLRepdf3YB+FIEXAMPLE=
x-mns-version: 2015-06-06
```

```
<?xml version="1.0" encoding="utf-8"?>
<Subscription xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Endpoint>http://company.com</Endpoint>
  <NotifyStrategy>BACKOFF_RETRY</NotifyStrategy>
  <NotifyContentFormat>SIMPLIFIED</NotifyContentFormat>
  <FilterTag>important</FilterTag>
</Subscription>
```

返回示例：

```
HTTP/1.1 201
x-mns-request-id:56667376B2B71C9C16000025
x-mns-version: 2015-06-06
Location: http://$AccountId.mns.cn-hangzhou.aliyuncs.com/topics/$TopicName/subscriptions/$SubscriptionName
```

## Special Error

错误代码	错误信息	状态码
SubscriptionNameLengthError	Subscription name length is out of range, should be between 1 and 255.	400

错误代码	错误信息	状态码
SubscriptionNameInvalid	The subscription you provided is invalid. SubscriptionName should start with alpha or digit, and contain only alpha, digit or -.	400
SubscriptionAlreadyExist	The subscription you want to create already exists.	409
EndpointInvalid	The endpoint you provided is invalid.	400
InvalidArgument	The length of filter tag should be between 1 and 16.	400

## 4.3.2. SetSubscriptionAttributes

使用SetSubscriptionAttributes接口修改Subscription的属性。

请求消息由以下几个部分组成：

- 请求行

```
PUT /topics/$TopicName/subscriptions/$SubscriptionName?metaoverride=true HTTP/1.1
```

- 特有URI参数

metaoverride=true 将会修改Subscription的属性。

- 特有 Request Header

无，请参见[公共参数](#)。

- Request Body

Request Body为XML格式，XML中包含修改Subscription的属性。

参数名称	是否必选	说明	参数值
NotifyStrategy	否	描述了向Endpoint推送消息出现错误时的重试策略。	BACKOFF_RETRY或者EXPONENTIAL_DECAY_RETRY，默认为BACKOFF_RETRY。重试策略的具体描述请参见 <a href="#">NotifyStrategy</a> 。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body  
无。

### 示例

请求示例：

```
PUT /topics/$TopicName/subscriptions/$SubscriptionName?metaoverride=true HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:09:22 GMT
Authorization: MNS 15B4D3461F177624206A:rNT4xclLS297/n0UFYlezzHJI7Q=
x-mns-version: 2015-06-06
<?xml version="1.0" encoding="utf-8"?>
<Subscription xmlns="http://mns.aliyuncs.com/doc/v1/">
  <NotifyStrategy>BACKOFF_RETRY</ NotifyStrategy>
</Subscription>
```

返回示例：

```
HTTP/1.1 204 No Content
x-mns-request-id:56667412B2B71C9C16000027
x-mns-version: 2015-06-06
```

### Special error

错误代码	错误消息	状态码
SubscriptionNotExist	The subscription you provided does not exists.	404

## 4.3.3. GetSubscriptionAttributes

使用GetSubscriptionAttributes接口来获取Subscription的属性。

### Request

请求消息由以下几个部分组成：

- 请求行

```
GET /topics/$TopicName/subscriptions/$SubscriptionName HTTP/1.1
```

- 特有 Request Header  
无，请参见公共参数。
- Request Body  
无。

## Response

返回消息由返回状态行、HTTP头和消息三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有 Response Header

无，请参见[公共参数](#)。

- Response Body

参数名称	说明
SubscriptionName	Subscription的名称。
Subscriber	Subscription订阅者的AccountId。
TopicOwner	Subscription订阅的主题所有者的AccountId。
TopicName	Subscription订阅的主题名称。
Endpoint	订阅的终端地址。
NotifyStrategy	向Endpoint推送消息错误时的重试策略。
NotifyContentFormat	向Endpoint推送的消息内容格式。
FilterTag	描述了该订阅中消息过滤的标签（仅标签一致的消息才会被推送）。
CreateTime	Subscription的创建时间，从1970-1-1 00:00:00到现在的秒值。
LastModifyTime	修改Subscription属性信息最近时间，从1970-1-1 00:00:00到现在的秒值。

## 示例

请求示例：

```
GET /topics/$TopicName/subscriptions/$SubscriptionName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:09:58 GMT
Authorization: MNS 15B4D3461F177624206A:GvaApggLQfMu6B6W5sbF6HEjV6Y=
x-mns-version: 2015-06-06
```

返回示例：

```

HTTP/1.1 200 OK
Content-Type=text/xml;utf-8
Content-Length:451
x-mns-request-id:56667436B2B71C9C16000029
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Subscription xmlns="http://mns.aliyuncs.com/doc/v1">
  <CreateTime>1449554806</CreateTime>
  <Endpoint>http://company.com</Endpoint>
  <FilterTag>important</FilterTag>
  <LastModifyTime>1449554962</LastModifyTime>
  <NotifyContentFormat>XML</NotifyContentFormat>
  <NotifyStrategy>BACKOFF_RETRY</NotifyStrategy>
  <SubscriptionName>$SubscriptionName</SubscriptionName>
  <Subscriber>$Subscriber</Subscriber>
  <TopicName>$TopicName</TopicName>
  <TopicOwner>$TopicOwner</TopicOwner>
</Subscription>
    
```

### Special error

错误码	错误信息	状态码
SubscriberNotExist	The subscription you provided does not exist.	404

## 4.3.4. Unsubscribe

使用Unsubscribe接口来取消一个已创建的Subscription。

### Request

Request的构造主要由以下几个部分组成：

- 请求行

```
DELETE /topics/$TopicName/subscriptions/$SubscriptionName HTTP/1.1
```

- 特有Request Header  
无，请参见[公共参数](#)。
- Request Body  
无。

### Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

```
HTTP/1.1 204 No Content
```

如果Unsubscribe的Subscription不存在，返回204，取消订阅成功。

- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

无。

## 示例

请求示例：

```
DELETE /topics/$TopicName/Subscriptions/$SubscriptionName HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:18:04 GMT
Authorization: MNS 15B4D3461F177624206A:UOEb7OmC9GX7G9BLY3XryP6vtjQ=
x-mns-version: 2015-06-06
```

返回示例：


```
HTTP/1.1 204 No Content
x-mns-request-id: 5666761CB2B71C9C16000035
x-mns-version: 2015-06-06
```

## 4.3.5. ListSubscriptionByTopic

使用ListSubscriptionByTopic接口列出某个主题下的Subscription列表，可分页获取数据。

### 描述

该接口用于列出某个主题下的Subscription列表，可分页获取数据。

 **说明** 如果本次请求结果返回的Subscription总数超过本次请求的 `x-mns-ret-number` 参数值，返回的XML结果中包含的 `NextMarker` 表示下一页开始位置，在下次请求时将该值赋予 `x-mns-marker` 参数以返回下一页的结果。

### Request

Request的构造主要由以下几个部分组成：

- 请求行

```
GET /topics/$TopicName/subscriptions HTTP/1.1
```

- 特有Request Header

参数名称	是否必选	说明
x-mns-prefix	否	按照该前缀开头的 <code>SubscriptionName</code> 进行查找。
x-mns-marker	否	请求下一个分页的开始位置，从上次分页结果返回的 <code>NextMarker</code> 获取。
x-mns-ret-number	否	单次请求结果的最大返回个数，1~1000范围内的整数，默认值为1000。

其它Request Header请参见[公共参数](#)。

- Request Body  
无。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 200 OK

- 特有Response Header  
无，请参见[公共参数](#)。

- Response Body

返回XML格式的结果，顶层 `Subscriptions` 元素下包含多个 `Subscription` 元素，`Subscription` 元素中包含 `SubscriptionURL`，`SubscriptionURL` 表示Subscription的URL访问地址。如果返回结果超过 `x-mns-ret-number` 的参数值，包含 `NextMarker` 元素表示下个分页的起始位置。

## 示例

请求示例：

```
GET /topics/$TopicName/subscriptions HTTP/1.1
x-mns-ret-number:2
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:16:18 GMT
Authorization: MNS 15B4D3461F177624206A:dXSLY5yGG8Ri5f26ZEmxMCq1CC4=
x-mns-version: 2015-06-06
```

返回示例：

```
HTTP/1.1 200 OK
x-mns-request-id:566675B2B2B71C9C16000033
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Subscriptions xmlns="http://mns.aliyuncs.com/doc/v1">
  <Subscription>
    <SubscriptionURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/$TopicName/subscriptions/$SubscriptionName1</SubscriptionURL>
  </Subscription>
  <Subscription>
    <SubscriptionURL>http://$AccountId.mns.cn-hangzhou.aliyuncs.com/$TopicName/subscriptions/$SubscriptionName2</SubscriptionURL>
  </Subscription>
  <NextMarker>OTczNjU4MTcvYmRwejd6NXluby8yNjg0Mi9+fn5FI1Rlc3RUb3BpYy0xI2JkcHo3ejV5bm8jVGVzdFN1Yi0zLzA= </NextMarker>
</Subscriptions>
```

## 4.4. 消息管理

### 4.4.1. PublishMessage

发布者使用PublishMessage接口来向指定的主题发布消息，消息发布到主题后随即会被推送给Endpoint消费。

#### Request

Request的构造主要由以下几个部分组成：

- 请求行

```
POST /topics/$TopicName/messages HTTP/1.1
```

- 特有Request Header

无，通用Header请参见[公共参数](#)，签名相关请参见[请求签名机制](#)。

- Request Body

Request Body为XML格式

如需推送邮件或短信，则MessageAttributes相应的设置说明如下：

- 如果需要推送到MailEndpoint，MessageAttributes需要包含DirectMail属性。

DirectMail属性的Text内容是JSON格式，需要包含以下字段，详情请参见[SingleSendMail](#)。



参数名称	是否必选	取值类型	说明
AccountName	是	String	发信账号。
Subject	是	String	邮件主题。
AddressType	是	Number	取值为0或1。
IsHtml	是	Number	邮件内容是否作为HTML处理，取值为0或1。
ReplyToAddress	是	Boolean	取值为0或1。

- 如果需要推送到SmsEndpoint，MessageAttributes需要包含DirectSMS属性。

DirectSMS属性的Text内容是JSON格式，需要包含以下字段：

参数名称	是否必选	说明
FreeSignName	是	短信签名。
TemplateCode	是	短信模板ID。
Type	是	singleContent或multiContent，表示单发或批量。如果取值为multiContent，则只会被Endpoint为sms:directsms:anonymous的订阅处理。
Receiver	否	如果Type为singleContent且Subscription的Endpoint是sms:directsms:anonymous，则此项生效且必填，填写接收人的手机号码，多个号码以逗号分隔。
SmsParams	是	JSON格式，一些具体参数。对于singleContent，此处填写格式为{"参数1": "Value1", "参数2": "Value2"}；对于multiContent，此处填写格式为{"电话号码1": {"参数1": "Value1", "参数2": "Value2"}, "电话号码2": {"参数1": "Value3", "参数2": "Value4"}}。

## Response

返回消息有返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code  
HTTP/1.1 201 Created
- 特有Response Header

无，请参见[公共参数](#)。

- Response Body

返回的结果为XML格式，包括MessageId和MessageBodyMD5：

参数名称	说明
MessageId	消息编号，在主题中唯一。
MessageBodyMD5	消息正文的MD5值。

## 示例

请求示例：

```
POST /topics/$TopicName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:13:40 GMT
Content-Length:500
Content-Type:text/xml;charset=utf-8
Authorization: MNS 15B4D3461F177624206A:aibzWu1iDEx9LwO56+kHgA3eqml=
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageBody>Base64 Encoded Result</MessageBody>
  <MessageTag>important</MessageTag>
  <MessageAttributes>
    <DirectMail>{"Subject":"TestMailSubject","AccountName":"direct_mail_account_name@aliyun-inc.com",
"ReplyToAddress":0,"AddressType":0,"IsHtml":0}</DirectMail>
  </MessageAttributes>
</Message>
```

批量发布短信消息请求示例：

```

POST /topics/$TopicName/messages HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Tue, 08 Dec 2015 06:13:40 GMT
Content-Length:500
Content-Type:text/xml;charset=utf-8
Authorization: MNS 15B4D3461F177624206A:aibzWu1iDEx9LwO56+kHgA3eqml=
x-mns-version: 2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageBody>content</MessageBody>
  <MessageAttributes>
    <DirectSMS>
      {
        "FreeSignName": "$value",
        "TemplateCode": "$value",
        "Type": "multiContent",
        "Receiver": "$num1,$num2",
        "SmsParams": "{$num1\":{ \"$key1\": \"$value1\", \"$key2\": \"$value2\"}, \"$num2\":{ \"$key1\": \"$value1\", \"$key2\": \"$value2\"}}"}
      }
    </DirectSMS>
  </MessageAttributes>
</Message>

```

返回示例：

```

HTTP/1.1 201 Created
Content-Length:120
Content-Type:text/xml;charset=utf-8
x-mns-request-id:56667514B2B71C9C1600002B
x-mns-version:2015-06-06

<?xml version="1.0" encoding="utf-8"?>
<Message xmlns="http://mns.aliyuncs.com/doc/v1/">
  <MessageId>D273CD6A89564E54-1-15180395A19-200000001</MessageId>
  <MessageBodyMD5>3AC6DD36D2D6B7283F2F490A09751BFE</MessageBodyMD5>
</Message>

```

## Special error

错误码	错误信息	状态码
TopicNotExist	The topic you provided does not exist.	404

## 4.5. 通知管理

### 4.5.1. HttpEndpoint

使用HttpEndpoint接口通过发送POST请求将Notifications推送到Endpoint。

#### 描述

消息服务MNS通过发送POST请求将Notifications推送到Endpoint端，Notifications支持两种格式：

- XML完整格式  
在Request Body中包含推送消息的正文和消息的Meta信息。
- SIMPLIFIED精简格式  
Request Body即为用户的消息正文，不做任何改动，MessageId会在Request Header中传递给用户。

#### Request

Request的构造由以下几个部分组成：

- 请求行

POST /notifications HTTP/1.1

默认是 /notifications，如果订阅者在创建Subscription时指定的Endpoint包含URI，会使用用户设置的URI替换 /notifications。

- 特有URI参数  
无。
- 特有Request Header

参数名称	说明	备注
Authorization	此次推送请求的签名	XML和SIMPLIFIED格式下均有
x-mns-signing-cert-url	签名证书的地址（Base64编码过）	XML和SIMPLIFIED格式下均有
x-mns-request-id	此次推送消息的Request编号	XML和SIMPLIFIED格式下均有
x-mns-message-id	此次推送消息的MessageId	SIMPLIFIED格式下有
x-mns-message-tag	此次推送消息的消息标签	SIMPLIFIED格式下有

- Request Body  
XML格式下，Request Body为XML格式，包含消息正文和Meta信息。

参数名称	说明
TopicOwner	被订阅主题的拥有者
TopicName	被订阅主题的名称
Subscriber	订阅者
SubscriptionName	订阅名称
MessageId	消息编号
Message	消息正文
MessageMD5	消息的MD5值
MessageTag	消息标签（用于消息过滤）
PublishTime	消息的发布时间，从1970-1-1 00:00:00 000到消息发布时的毫秒值。

精简格式下，Request Body为发布者发布的消息正文。

## Response

返回消息由返回状态行、HTTP头和消息体三部分组成：

- HTTP Status Code

HTTP/1.1 204 No Content

- 正常处理通知消息，返回204；超过5秒未返回204，则认为超时，对该消息重试推送。
- 请求签名验证不通过，返回403。
- 其他任何错误，返回500。

- 特有Response Header

无。

- Response Body

无。

## 示例

请求示例：

- XML 格式

```
POST /notifications HTTP/1.1
Host: company.com
Date: Tue, 08 Dec 2015 06:42:00 GMT
Content-Length: 300
Content-Type: text/xml;charset=utf-8
Content-MD5: OGQzNjAwYjIwMWFiMTliZDlhNzc4YjAwZjc5MGQ4NDE=
Authorization: QSNS+vn3lqJI4lOh4X/dxTv7Qebi/wrEaBJlzFFTc0hJVRUmrHLQo24ojBUETpz48+jrsjxMGP
JTILDw752Jww==
User-Agent: Aliyun Notification Service Agent
x-mns-request-id: 56667BB81FD0E7D1200024A0
x-mns-version: 2015-06-06
x-mns-signing-cert-url: aHR0cDovL2luc3Rlc3Qub3NzLWVuLWVhbmd6aG91LmFsaXl1bmNzLmNvbS94NT
A5X3B1YmXpY19jZXJ0aWZpY2F0ZS5wZW0=

<?xml version="1.0" encoding="utf-8"?>
<Notification xmlns="http://mns.aliyuncs.com/doc/v1/">
  <TopicOwner>$TopicOwner</TopicOwner>
  <TopicName>$TopicName</TopicName>
  <Subscriber>$Subscriber</Subscriber>
  <SubscriptionName>$SubscriptionName</SubscriptionName>
  <MessageId>6CC4D900CA59A2CD-1-15180534A8F-200000002</MessageId>
  <Message>{1:"a", 2:"b"}</Message>
  <MessageMD5>F1E92841751D795AB325861034B5CB55</MessageMD5>
  <MessageTag>important</MessageTag>
  <PublishTime>1449556920975</MessagePublishTime>
</Notification>
```

- SIMPLIFIED 格式

```
POST /notifications HTTP/1.1
Host: company.com
Date: Tue, 15 Dec 2015 09:12:46 GMT
Content-Length: 487
Content-Type: text/plain;charset=utf-8
Content-MD5: NmY4Njk0MzlmNzg3NTFkMTJiN2M5OWE4ZmYzNTU4ZWQ=
Authorization: aXZlpRkRmlavmwceCyrPatB7SaEbOJdWpmFWYX3nFwtmaixK/M/3kl1njY0pgM7mrIB7NIB
v5NfUZnkcjlO4A==
User-Agent: Aliyun Notification Service Agent
x-mns-message-id: 6CC4D900CA59A2CD-1-15180534A8F-200000002
x-mns-message-tag: important
x-mns-request-id: 566FD963B2B71C44481F701A
x-mns-signing-cert-url: aHR0cDovL21uc3Rlc3Qub3NzLWNuLWVhbmmd6aG91LmFsaXl1bmNzLmNvbS94NT
A5X3B1YmxpY19jZXJ0aWZpY2F0ZS5wZW0=
x-mns-version: 2015-06-06

{"1:"a", 2:"b"}
```

返回示例：

```
HTTP/1.1 204 No Content
```

## 5. 错误响应

### 5.1. 错误响应格式

当用户访问MNS出错时，MNS会返回给用户一个合适的3xx、4xx或5xx的HTTP状态码，以及一个text或xml格式的消息体。

错误响应的消息体例子：

```
<?xml version="1.0" encoding="UTF-8"?>
<Error xmlns="http://mns.aliyuncs.com/doc/v1/">
  <Code>InvalidArgument</Code>
  <Message>Authorization header is invalid or missing.</Message>
  <RequestId>04bb60e3-28aa-4706-99b0-9345d834f266</RequestId>
  <HostId>mns.cn-hangzhou.aliyuncs.com</HostId>
</Error>
```

所有错误的消息体中都包括以下几个元素：

- Code：MNS返回给用户的错误码。
- Message：MNS给出的详细错误信息。
- RequestId：用于唯一标识该次请求的编号。  
当您无法解决问题时，可以提供这个RequestId寻求MNS支持工程师的帮助。
- HostId：用于标识访问的MNS服务的地域。

其他特殊的错误信息元素请参见每个请求的具体介绍。

### 5.2. 错误码

本文介绍了错误响应中错误码的错误描述和建议处理方式。

错误码	错误描述	HTTP状态码	建议处理方式
AccessDenied	The OwnerId that your Access Key Id associated to is forbidden for this operation.	403	检查AccessKeyId和AccessKeySecret等配置是否正确。
InvalidAccessKeyId	The AccessKey Id you provided is not exist.	403	检查AccessKeyId是否正确。
InternalServerError	Internal error.	500	系统错误，请联系MNS技术支持。
InternalServerError	Internal error.	500	系统错误，请联系MNS技术支持。



错误码	错误描述	HTTP状态码	建议处理方式
InvalidAuthorizationHeader	The Authorization header format is invalid.	400	签名格式不正确，请参见 <a href="#">请求签名机制</a> 。
InvalidDateHeader	The Date header format is invalid.	400	Date字段不合法，请参见 <a href="#">公共参数</a> 。
InvalidArgument	The XML you provided did not validate against our published schema, cause by Element①.	400	XML结构不正确，请参见对应API文档。
InvalidArgument	The value of Element② should between Low③ and High④ seconds/bytes.	400	参数值非法，请根据提示调整。
InvalidDigest	The Content-MD5 you specified is invalid.	400	请求Header中Content-MD5不正确。
InvalidRequestURL	Http request URL format invalid.	400	请求的URL不正确，请参见对应API文档。
InvalidQueryString	Http request URL contains invalid querystring item "Element⑤" .	400	请求的URL不正确，请参见对应API文档。
MalformedXML	The XML you provided was not well-formed.	400	XML结构异常，请参见对应API文档。
MissingAuthorizationHeader	Authorization header is required.	400	请求Header缺少字段，请参见 <a href="#">公共参数</a> 。
MissingDateHeader	Date header is required.	400	请求Header缺少字段，请参见 <a href="#">公共参数</a> 。
MissingReceiptHandle	ReceiptHandle is required.	400	请求缺少参数，请参见对应API文档。
MissingVisibilityTimeout	VisibilityTimeout is required.	400	请求缺少参数，请参见对应API文档。
MessageNotExist	Message not exist.	404	队列中没有可见消息。
	The receipt handle you provided has expired.	404	消费消息过慢导致消息重新回到队列生成新的ReceiptHandle，之前的ReceiptHandle失效。加快消费速度或者增加队列的不可见时间。

错误码	错误描述	HTTP状态码	建议处理方式
QueueAlreadyExist	The queue you want to create is already exist.	409	同名队列已存在，且请求属性与原有队列属性不同。
InvalidQueueName	The queue name you provided is invalid. QueueName should start with alpha and contain only alpha, digit or - .	400	队列名称不合法，请参见对应API文档。
QueueNameLengthError	Queue name length should between 1 and 255.	400	队列名称不合法，请参见对应API文档。
QueueNotExist	The queue name you provided is not exist.	404	队列不存在，请先创建队列。
ReceiptHandleError	The receipt handle you provide is not valid.	400	请求参数不合法，请参见对应API文档。
SignatureDoesNotMatch	The request signature we calculated does not match the signature you provided. Check your key and signing method.	403	请求签名异常，请参见 <a href="#">请求签名机制</a> 。
TimeExpired	The http request you sent is expired.	408	请求时间与MNS服务器时间相差超过15分钟，建议检查本地时间。
QpsLimitExceeded	The qps limit of the queue is exceeded ⑥.	400	队列每秒的请求次数超过QPS限制，如果需要提高QPS限制，请联系MNS技术支持。
TopicAlreadyExist	The topic you want to create already exists.	409	同名主题已存在，且请求属性与原有主题属性不同。
TopicNameInvalid	The topic name you provided is invalid. TopicName should start with alpha or digit and contain only alpha, digit or - .	400	主题名称不合法，请参见对应API文档。
TopicNameLengthError	Topic name length is out of range, should be between 1 and 255.	400	主题名称不合法，请参见对应API文档。

错误码	错误描述	HTTP状态码	建议处理方式
TopicNotExist	The Topic you provided does not exist.	404	主题不存在，请先创建主题。
SubscriptionNameInvalid	The subscription name you provided is invalid. SubscriptionName should start with alpha or digit and contain only alpha, digit or -.	400	订阅名称不合法，请参见对应API文档。
SubscriptionNameLengthError	Subscription name length is out of range, should be between 1 and 255.	400	订阅名称不合法，请参见对应API文档。
SubscriptionNotExist	The subscription you provided does not exist.	404	订阅不存在，请先创建订阅。
SubscriptionAlreadyExist	The subscription you want to create already exists.	409	同名订阅已经存在，且请求属性与原有订阅属性不同。
EndpointInvalid	The format of endpoint you provided is invalid.	400	Endpoint参数不合法，请参见Endpoint。
InvalidArgument	The length of message should not be larger than MaximumMessageSize.	400	消息体过长，请参见对应API文档。

#### ② 说明

- ①②：在请求内容的XML元素名称。
- ③：在MNS中某个参数的下限值。
- ④：在MNS中某个参数的上限值。
- ⑤：在URL请求中QueryString的元素。
- ⑥：当前MNS单个queue的QPS上限是5 KB。

## 6.API问题

### 6.1. API使用说明

本文介绍消息服务MNS API的使用概况。

消息服务MNS提供了以下API接口：

- **队列接口**

队列接口适用于点对点的消息收发，当接收消息时，需要应用端自行轮询获取消息（拉模式）。

- **主题接口**

主题接口适用于一对多的消息收发，应用端只需要在某个地址上启动监听，服务端就会主动将消息推送过去（推模式）。

如果问题未能解决，请联系[售后技术支持](#)。

### 6.2. 为什么我的签名总是没算对？

本文结合实例对签名方法进行说明。

有关签名计算的原理以及相关注意事项请参见[API参考](#)。结合实例对签名方法进行说明，请求的HTTP header参数为：

```
GET /MyQueue HTTP/1.1
Host: $AccountId.mns.cn-hangzhou.aliyuncs.com
Date: Thu, 09 Jul 2015 03:01:34 GMT
x-mns-version:2015-06-06
```

需要进行加密的签名源字符串为：

```
GET
(换行符)
(换行符)
Thu, 09 Jul 2015 03:01:34 GMT
x-mns-version:2015-06-06
/MyQueue
```

假设accessId为TestAccessID，accessKey为TestAccessSecret，经过加密算法得出的签名值为uwX3yeWoiLzgmvesW0BQsGfM7b8=。