

阿里云 消息服务MNS

常见问题

文档版本：20200507

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. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能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。
	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。
	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。	执行cd /d C:/window命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid Instance_ID
[]或者[a b]	表示可选项，至多选择一个。	ipconfig [-all -t]
{ }或者{a b}	表示必选项，至多选择一个。	switch {active stand}

目录

法律声明.....	I
通用约定.....	I
1 MNS常见问题.....	1
2 消息通知常见问题.....	8
3 MNS费用查询.....	10
4 技术分享.....	11
4.1 MNS API使用说明.....	11
4.2 MNS SDK下载.....	11
4.3 MNS主题订阅的使用限制.....	11
4.4 MNS删除消息时返回message not exist.....	12
4.5 MNS返回Http request URL format invalid.....	13
4.6 MNS The OwnerId that your Access Key Id associated to is forbidden原因分析....	14
4.7 MNS签名报错403, SignatureDoesNotMatch.....	16
4.8 MNS消费端代码如何实现消息监听.....	19

1 MNS常见问题

本文介绍MNS的基本概念、使用相关的常见问题。

什么是MNS？

- MNS是Message Notification Service的简称，是阿里云提供的一种高效、可靠、安全、便捷、可弹性扩展的分布式消息服务。
- MNS能够帮助应用开发者在应用的分布式组件上自由的传递数据，构建松耦合系统。

用MNS可以做什么？

典型的使用场景：

1. 将MNS与其他阿里云产品集成，让应用程序更加可靠、灵活。
2. 将MNS用作工作队列，其中每条消息代表一项任务，需要通过一个流程来完成。一台或多台ECS可以从队列中读取并执行任务。
3. 将业务流程中重要事件的通知保存在MNS中，每个事件在队列中都有一条对应的消息，需要知晓该事件的应用程序可以读取和处理对应的消息。

阿里云各地域对应关系说明

随着阿里云业务的不断发展，产品的地域也不断丰富，为了规范命名和适应国际化发展，阿里云将于2016年3月29日对各云产品地域进行更名，新旧名称对应关系如下表：

原名称	新的名称	英文名称
青岛	华北1	China North 1
北京	华北2	China North 2
杭州	华东1	China East 1
上海	华东2	China East 2
深圳	华南1	China South 1

MNS不同地域的接入地址是什么？

区域	公网访问地址	私网访问地址	VPC私网访问地址
华北1	http(s)://{AccountId}.mns.cn-qingdao.aliyuncs.com	http://{AccountId}.mns.cn-qingdao-internal.aliyuncs.com	http://{AccountId}.mns.cn-qingdao-internal-vpc.aliyuncs.com

区域	公网访问地址	私网访问地址	VPC私网访问地址
华北2	http(s)://{AccountId}.mns.cn-beijing.aliyuncs.com	http://{AccountId}.mns.cn-beijing-internal.aliyuncs.com	http://{AccountId}.mns.cn-beijing-internal-vpc.aliyuncs.com
华东1	http(s)://{AccountId}.mns.cn-hangzhou.aliyuncs.com	http://{AccountId}.mns.cn-hangzhou-internal.aliyuncs.com	http://{AccountId}.mns.cn-hangzhou-internal-vpc.aliyuncs.com
华东2	http(s)://{AccountId}.mns.cn-shanghai.aliyuncs.com	http://{AccountId}.mns.cn-shanghai-internal.aliyuncs.com	http://{AccountId}.mns.cn-shanghai-internal-vpc.aliyuncs.com
华南1	http(s)://{AccountId}.mns.cn-shenzhen.aliyuncs.com	http://{AccountId}.mns.cn-shenzhen-internal.aliyuncs.com	http://{AccountId}.mns.cn-shenzhen-internal-vpc.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	http://{AccountId}.mns.ap-southeast-1-internal-vpc.aliyuncs.com
美国西部1（硅谷）	http(s)://{AccountId}.mns.us-west-1.aliyuncs.com	http://{AccountId}.mns.us-west-1-internal.aliyuncs.com	http://{AccountId}.mns.us-west-1-internal-vpc.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	无
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	无
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	无

区域	公网访问地址	私网访问地址	VPC私网访问地址
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	无
华东1金融云	无	http://{AccountId}.mns.cn-hzjbp-a-internal.aliyuncs.com	无
华东2金融云	无	http://{AccountId}.mns.cn-shanghai-finance-internal.aliyuncs.com	无
华南1金融云	无	http://{AccountId}.mns.cn-shenzhen-finance-internal.aliyuncs.com	http://{AccountId}.mns.cn-shenzhen-finance-internal-vpc.aliyuncs.com

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。

Account ID是什么？

Account ID在阿里云注册用户的账号ID，可在阿里云官网查看。

MNS提供哪些可用的操作？

MNS队列模型支持的操作指令如下所示。

接口	功能描述	支持方式
#unique_4	创建队列	控制台、API调用
#unique_5	设置队列属性	控制台、API调用
#unique_6	获取队列属性	控制台、API调用
#unique_7	删除队列	控制台、API调用
#unique_8	获取队列列表	控制台、API调用

接口	功能描述	支持方式
#unique_9	发送消息	控制台、API调用
#unique_10	批量发送消息	API调用
#unique_11	消费消息	控制台、API调用

接口	功能描述	支持方式
#unique_12	批量消费消息	API调用
#unique_13	删除消息	控制台、API调用
#unique_14	批量删除消息	API调用
#unique_15	查看消息	API调用
#unique_16	批量查看消息	API调用
#unique_17	修改消息下次可消费时间	API调用

MNS主题模型支持的操作指令如下所示。

接口	功能描述	支持方式
#unique_18	创建主题	控制台、API调用
#unique_19	修改主题属性	控制台、API调用
#unique_20	获取主题属性	控制台、API调用
#unique_21	删除主题	控制台、API调用
#unique_22	获取主题列表	控制台、API调用

接口	功能描述	支持方式
#unique_23	创建订阅	控制台、API调用
#unique_24	修改订阅属性	控制台、API调用
#unique_25	获取订阅属性	控制台、API调用
#unique_26	删除订阅	控制台、API调用
#unique_27	获取主题的订阅列表	控制台、API调用

接口	功能描述	支持方式
#unique_28	发布消息	控制台、API调用

MNS相对于自主研发、商用或开源的消息队列系统有哪些好处？

与自行构建系统来管理消息队列或使用商用、开源的消息和通知服务相比，使用MNS的好处是：

- 无需大量前期投入来进行开发、配置；
- 随着您业务量的不断增大，无需持续投入硬件和管理资源；
- 缺省提供消息冗余存储，以便在硬件发生故障时不会丢失消息，相应的系统投入、开发、配置和部署将更为简单；

- 无需为消息服务后期投入部署和维护的资源，只需简单配置即可用于生产环境。

如何使用MNS？

1. 注册阿里云账号并在线开通MNS服务。
2. 通过管理控制台的**账号管理** 获取帐号ID，单击AccessKeys访问密钥（AccessKeyID）。
3. 进入消息服务管理控制台可以体验可视化的队列创建和删除、收发消息等基本操作。
4. 在应用程序中调用API接口（SDK）执行MNS的所有操作。

系统中如何标识消息？

- MessageId用来标识在队列/主题中的一个消息，在一个队列/主题中每个消息都有唯一的MessageId，但在不同的队列/主题之间中并不唯一。
- 当消息发送到MNS队列/主题时，MNS会生成一个MessageId，此ID一旦产生就不会改变。
- 在队列模式下，当消息被取出时，MNS会将消息的消息体、MessageId以及该次请求的临时句柄（ReceiptHandle）一齐返回给用户。在有效期内，用户消费完毕之后，使用ReceiptHandle进行消息的删除。

MNS是否支持长轮询？

支持。与传统的短轮询相比，长轮询只会在消息进入队列或长轮询超时时才返回响应。一旦消息可用，长轮询可立即以简单经济的方式从您的MNS队列检索消息。

有关长轮询的具体设置，请参见MNS [API参考](#)中有关PollingWaitSeconds属性的说明。

MNS是否提供对消息的先入先出（FIFO）访问？

MNS消费消息时尽量做到先进先出，正是因为分布式消息队列的一些特性并不能保证您能按照消息的发送顺序消费消息，如果您的业务必需先进先出，建议在消息中加入序号信息以便消费消息后进行重新排序。

MNS是否可以与其他阿里云产品协作使用？

MNS可以与ECS、OSS和OTS等阿里云服务一起使用，从而让应用程序具有更好的灵活性和可扩展性。常见的使用案例包括：创建需要互相通信却又不能同时处理相同工作量的多个组件或模块。在此情形中，MNS队列可以承载消息，让用户在ECS实例上运行的应用程序能够有序地进行处理。ECS实例可以读取该队列，处理任务，然后将结果作为消息发布到另一MNS队列（可能由其他应用程序进行进一步处理）。由于ECS允许应用程序动态扩展，应用程序开发者可根据MNS队列中的消息数（工作量），轻松改变计算实例的数量，确保作业得以及时处理。

如何确保在MNS中存储的数据的可靠性？

MNS将所有队列和消息存储在阿里云高可靠、高可用的数据中心组成的网络中。所有消息冗余地存储在多台服务器上，当某台服务器出现问题时，冗余数据将会自动拷贝至其他服务器；这意味着单台服务器不可用或网络故障都不会对MNS队列中消息的安全性造成任何影响。

MNS如何保证多个消息的消费者访问同一消息队列，而不会丢失消息或重复消费消息？

每个MNS队列都具有可配置的不可见时间段属性（即队列的取出消息隐藏时长属性）。当队列中的某条消息被取出后，在不可见时间段内，其他消费者将无法获取到该条消息。如果用户在不可见时间段内完成了消息的消费，则需要通过临时句柄（ReceiptHandle）进行消息的删除；如果超过不可见时间仍未完成消费，则需要发送不可见时间段的延长请求（ChangeVisibilityTimeout）；否则消息将会在不可见时间段结束后重新被其他消费者取出。

每条消息会被接收多少次？

我们会在系统设计上保证您队列中的所有消息至少被消费一次，建议您加强应用服务的容错性，以便多次处理同一消息时不会造成错误或不一致。

如何保证我的消息队列中的消息是安全的？

我们提供了安全可靠的身份验证机制，确保您的MNS队列受到保护，防止遭到未经授权的访问。只有阿里云账户拥有者才能访问其创建的队列。

如何配置MNS以支持更长的消息保留期？

要配置消息保留期限，请使用SetQueueAttributes方法设置MessageRetentionPeriod属性。此属性用于指定消息在MNS队列中保留的秒数。目前，消息保留期限的默认值为4天。使用MessageRetentionPeriod属性时，消息保留期限可以设置为60秒（1分钟）到最长604800秒（7天）之间的任何值。

消息在MNS中可以保留多久？

MNS中的消息保留期限是可配置的，您可以设置为1分钟到7天之间的任何值。默认值为3天；一旦达到消息保留期限，您的消息会被自动删除。较长消息保留期选项可提供更大的灵活性，允许在消息产生和消耗之间存在更长的间隔。

如何配置MNS以支持更大的消息大小？

要配置最大消息大小，请使用SetQueueAttributes方法设置MaximumMessageSize属性。此属性指定MNS队列中的消息可以包含的字节数限值。可以设置为1024字节（1KB）到最大65536字节（64KB）之间的任何值。如果消息长度超过了64KB，建议将数据存储在OSS中，MNS只存储数据访问地址。

对于较长时间没有任何消息的队列，会怎么处理？

在您正常使用MNS服务期间，阿里云不会对非活跃的队列/主题进行删除操作。但是，如果您的MNS使用权限因为欠费或其他原因被停止了，那么我们将删除您名下的所有队列和主题。

为什么我的签名总是没算对？

有关签名计算的原理以及相关注意事项请参见MNS API参考手册，这里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uwX3yeWolLzgmvesW0BQSgfM7b8=。

不同地域中的同名队列是同一个队列吗？

不是，不同地域的同名队列是各自独立的。

删除队列实例之后，是否仍然会继续计费？

如果您不再需要使用某个队列，在删除该队列实例之后，同时需要停止针对该队列的所有API请求，否则MNS仍然会按照API请求次数进行计费。

删除主题实例之后，是否仍然会继续计费？

如果您不再需要使用某个主题，在删除该主题实例之后，从删除的第二天起将不再收取该主题的实例占用费，同时需要停止针对该主题的所有API请求，否则MNS仍然会按照API请求次数进行计费。

2 消息通知常见问题

本文介绍在使用MNS消息通知功能中的常见问题。

1. HttpEndpoint接收不到消息怎么办？如何进行快速调试？

- 强烈建议使用[#unique_31](#)进行调试，可确定是否是HttpEndpoint接收消息的逻辑有问题。使用辅助开发工具可分别针对鉴权逻辑以及处理消息的逻辑进行调试。
- 如果通过辅助开发工具确认HttpEndpoint可以正常接收推送请求，但仍然无法接收线上服务的推送消息，建议使用[#unique_32](#)，能够获取消息服务接收发布的消息以及推送消息所有相关的日志，可通过日志查看消息服务的推送过程以及结果。如果通过日志无法排查问题，可提交[工单](#)处理。

2. 如何确认推送请求中x-mns-signing-cert-url所标识的证书地址是MNS官方给出的？

- 确认这点后才能保证推送请求是MNS发出的。
- 目前MNS是将公钥证书放到OSS的固定Bucket（mnstest）中的，地址为：`https://mnstest.oss-cn-hangzhou.aliyuncs.com/x509_public_certificate.pem`，考虑到安全性，证书会定期更换，文件名也会发生变化，但一定还是放在Bucket（mnstest）中的。所以在判断证书地址的有效性时可通过前缀匹配的方式进行，即判断证书地址的前缀是否为：`https://mnstest.oss-cn-hangzhou.aliyuncs.com/`即可。
- 将来可能会考虑将证书放到其他更安全的地方，届时会通知大家。

3. HttpEndpoint端验权时每次远程请求公钥证书会比较耗时，如果提高性能？

- 强烈建议根据公钥证书地址，按Key-Value的方式缓存证书，这样就可以从内存中直接获取证书。MNS的每个推送请求中都会携带公钥证书地址，如果证书更换，那对应的地址也会更新（文件名会更新，例如加上版本号），只要地址没有更新，就可以一直拿缓存的证书进行鉴权。所以在处理推送请求时，可以根据x-mns-signing-cert-url描述的地址从缓存中直接获取证书，获取不到再远程访问，并存到缓存中以便下次使用。
- 例如：证书地址为：`https://mnstest.oss-cn-hangzhou.aliyuncs.com/x509_public_certificate.pem`，最简单的就可以使用如下方式缓存：

```
typedef map<string, string> cache;  
cache["x509_public_certificate.pem"] = "$content";  
cache["x509_public_certificate.pem.version2"] = "$content2";
```

4. HttpEndpoint签名认证失败，如何调查？

用户HttpEndpoint使用了某些HttpServer框架，对HttpHeader的Key进行了格式转化，导致用户在签名认证的时候拿到的MNS自定义的HttpHeader的Key已经发生了变化（注：MNS自定义的Header的Key是**全小写**的），最常见的是变成了首字母大写。针对这种情况，只需要在认证之前，将HttpHeader的Key全部转为小写就可以了。例如：

```
POST
4448CB3A3FCF8389296C49467F27E1D6
text/xml; charset=utf-8
Thu, 16 Jun 2016 04:52:04 GMT
X-Mns-Message-Id:A2657F346FADF895-1-155578D466D-20000164E
X-Mns-Request-Id:57623074628ECE0F5521DD4C
X-Mns-Signing-Cert-Url:aHR0cDovL21uc3Rlc3Qub3NzLWVuLWVhbmdd6aG91LmFsaX11bmNzLmNvbS94NTA5X3B1YmtpY19jZXJ0aWZpY2F0ZS5wZW0=
X-Mns-Version:2015-06-06
```

5. 订阅成功后，没有收到消息，并且日志中也查不到相关记录，怎么查？

这里列举下可能的一些问题：

确认下在订阅时，是否制定了FilterTag，如果制定了，在PublshMessage时，不指定MessageTag的话，该消息是会被过滤掉的，从而不会被推送（消息日志中也查不到该记录）。

6. 使用PHP编写HttpEndpoint时接收不到消息体，怎么办？

确认下您的处理逻辑中，是否是直接使用\$POST来获取HttpBody的，如果是的话，需要改成这种方式：

```
file_get_contents("php://input");
```

原因：最常见的\$_POST['fieldname']方法，只能接收Content-Type: application/x-www-form-urlencoded提交的数据，也就是表单POST过来的数据。而MNS推送请求中的Content-Type的类型仅支持：

```
text/xml; charset=utf-8
text/plain; charset=utf-8
application/json; charset=utf-8
```

file_get_contents("php://input")这种方法允许读取POST的原始数据，对于不同Content-Type的POST数据都可以使用，甚至对于未指定Type的，也可以用来获取原始数据。所以建议使用这种方法来处理推送请求。

3 MNS费用查询

MNS已于2015年9月15日正式商用，并于2015年9月24日正式收费。为便于您评估使用费用，您可以按如下方式查询消费明细：

1. 访问官网 [费用中心](#)。
2. 在左侧导航栏，选择**消费记录 > 消费明细**。在产品筛选栏中选择**消息服务**，并取消**隐藏0元消费明细**的筛选条件进行查询。
3. 通过点击查询返回记录中的**详情** 查看账单详细信息，您可以看到MNS的实际使用量。

如果问题还未能解决，请联系[售后技术支持](#)。

4 技术分享

4.1 MNS API使用说明

本文介绍MNS API的使用概况。

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）。

主题接口中的主题类似于JMS中的主题概念，详情请参见[MNS支持JMS协议的方案](#)。

如果问题未能解决，请联系[售后技术支持](#)。

4.2 MNS SDK下载

MNS提供了以下语言版本SDK供您使用，一般建议下载最新发布的版本以获得最佳性能和稳定性。

- 官方下载：

- [Java SDK](#)
- [Python SDK](#)
- [C# SDK](#)
- [PHP SDK](#)
- [#unique_43](#)

- [非官方SDK](#)：C# SDK、PHP SDK、Node.js SDK等

如果问题还未能解决，请[提交工单](#)联系售后技术支持。

4.3 MNS主题订阅的使用限制

本文介绍MNS主题订阅功能的使用限制。

- 地域限制

阿联酋（迪拜）不支持主题。

- 主题个数

主题个数没有上限，但是一个主题下的sub个数上限为100。

- 推送限制

主题消息推送不能指定到sub。

发送消息时，目前只能指定是某个队列或者是某个主题，前者使用[队列接口](#)，后者使用[主题接口](#)。

如果需要专门发送消息给某一个应用端，那么可以使用队列接口，单独给应用端创建专门的队列，那么这个队列的消息就只有这个应用端能消费。

后续主题有计划支持将消息推送到指定的特定sub。

- 如果本地域不支持主题，又想使用主题服务，该怎么办？

例如，您的ECS在阿联酋（迪拜），但是阿联酋（迪拜）不支持主题，那么您可以在支持主题的印度（孟买）创建好主题后，使用主题的公网访问地址，同时，阿联酋（迪拜）的ECS需要开通公网。

使用这种方式，相对于内网环境，网络延迟是会大一些，但是作为前期调试的一种手段（需要在本地调试程序），也是可取的。

如果问题还未能解决，请联系[售后技术支持](#)。

4.4 MNS删除消息时返回message not exist

本文介绍了当MNS删除消息时返回message not exist的问题描述、问题原因和解决方案。

问题描述

删除消息时，如果返回message not exist，那么说明此消息ReceiptHandle已经失效。

问题分析

当消息被消费端接收到，消息状态将从active变为inactive，此inactive状态将持续Visibility Timeout的时间，在这段时间里，您可以删除这条消息；VisibilityTimeout一旦时间到，消息状态将从inactive变回为active，消息可继续被其他消费端消费，此时旧的消息ReceiptHandle失效，如果拿着旧的ReceiptHandle去删除消息，那么系统将返回message not exist。

- 如果VisibilityTimeout设置得太小，来不及删除消息，会导致message not exist;
- 但是如果设置得太大，一旦用户消息处理失败，又无法快速再次处理这条消息了。

如果消息处理失败，想快速再次处理，又不想因为VisibilityTimeout太小导致消息频繁超时，可参见[最佳实践：如何使用MNS的ChangeMessageVisibility](#)。

解决方案

检查队列属性VisibilityTimeout大小，设置一个合适的值，并等下次接收到消息时，再删除消息。

如果问题还未能解决，联系[售后技术支持](#)。

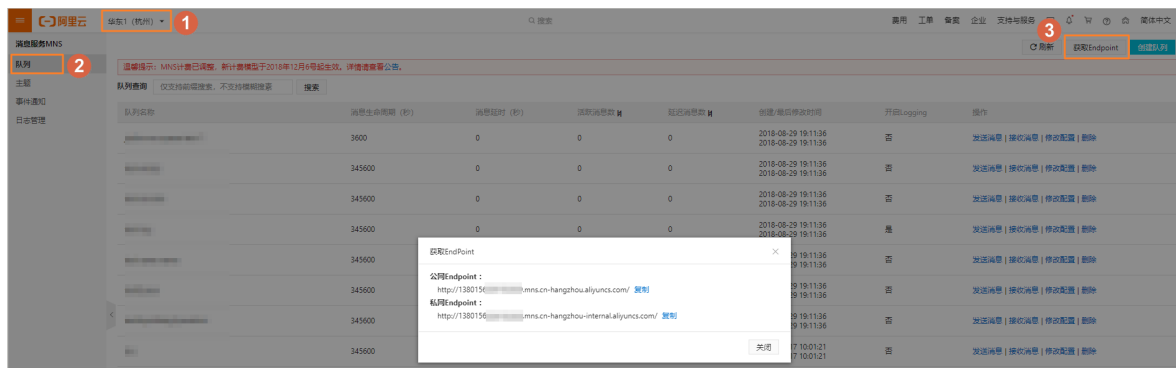
4.5 MNS返回Http request URL format invalid

当用户发出请求的URL有误时，MNS服务器端会返回Http request URL format invalid。本文介绍不同情况下如何排查。

- 如果使用SDK, 请检查代码中endpoint值。

```
CloudAccount account = new CloudAccount(<ak>,<sk>,<endpoint>);
```

在MNS控制台执行如下操作，即可获取Endpoint。



说明：

不少用户在代码里将Endpoint写成了队列地址，结果导致URL format invalid错误，例如：

- （正确）String endpoint = "http://44404.mns.cn-beijing.aliyuncs.com/";
- （错误）String endpoint = "http://44404.mns.cn-beijing.aliyuncs.com/queues/LTQueue/";

- 如果使用队列API, 请检查使用的是新协议还是历史协议，并根据协议检查url是否符合规范
header使用x-mns-version:2015-06-06，服务端将认为使用[新协议](#)。

header使用x-mqs-version:2014-07-08，服务端将认为使用[历史协议](#)。

新旧协议的URL规范不同，如下表所示。

协议版本	CreateQueue	SendMessage
新协议 (x-mns-version: 2015-06-06)	PUT /queues/\$queueName	POST /queues/\$queueName/messages

历史协议（x-mqs-version: 2014-07-08）	PUT /\$queueName	POST /\$queueName/messages
---------------------------------	------------------	----------------------------

**说明：**

您需严格按照协议规范来发送请求。

- 如果使用订阅API, 请检查host部分是否为支持主题的地域。

目前仅阿联酋（迪拜）不支持主题。

如果问题还未能解决, 请联系[售后技术支持](#)。

4.6 MNS The OwnerId that your Access Key Id associated to is forbidden原因分析

调用MNS SDK或者API, 如果出现“403 AccessDenied”或“The OwnerId that your Access Key Id associated to is forbidden for this operation”错误时, 一般是因为跨域访问队列, 或者是使用了错误的OwnerId, 或者是没有正确配置主账号RAM信息（使用子账号访问）。

什么是跨域访问？

如果使用私网队列地址, 那么生产端、消费端必须部署在相应地域的ECS上, 否则, 会报上述的错误。

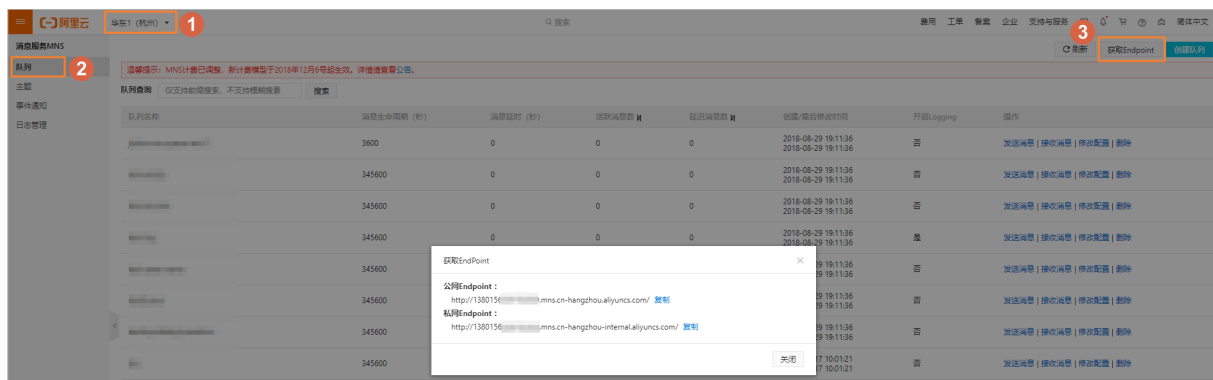
如果使用公网队列地址, 那么生产端、消费端可以部署在本地, 也可以部署在任意地域的ECS上（前提是ECS开通了公网, 可访问外网）。

什么是OwnerId？

OwnerId就是用户的阿里云账号ID, 用户可以在控制台选择**账号管理 > 安全设置**即可查看。



MNS队列成功创建后, 单击**获取Endpoint**, 也可查看以账号ID为前缀的私网Endpoint和公网Endpoint。



如果OwnerId确实没用错，但是还是出现了同样的报错信息，这又是为什么呢？

MNS在正式商用前，名为MQS，用户会发现正式商用前后，获取地址中显示的访问地址是不同的，例如：

- 正式商用前，旧地址：<http://y9g.mqs-cn-beijing.aliyuncs.com/>
- 正式商用后，新地址：<http://44404.mns.cn-beijing.aliyuncs.com/>

（后面篇幅把<http://y9g.mqs-cn-beijing.aliyuncs.com>叫旧域名，<http://44404.mns.cn-beijing.aliyuncs.com>叫新域名）

MNS正式商用后，在兼容旧协议规则的同时，启用了新协议规则，http header x-mns-version就是服务端用来识别新旧协议的标志头。

- 如果用户发送的请求带x-mns-version: 2015-06-06，那么服务端将适用新协议规则解析。
- 如果用户发送的请求带x-mqs-version: xxxx-xx-xx，那么服务端将适用旧协议规则解析。

所以当调用HTTP API时，使用的URL地址必须和标志头对应。

如果URL使用新地址，标志头却用x-mqs-version: xxxx-xx-xx，那么还是会报错。

以上是针对使用HTTP API用户来说的。

对于使用SDK的用户来说，如果使用官方提供的SDK，是不会出现这个问题的。一般出现问题的，都是使用非官方SDK的用户。非官方SDK一般没做新旧域名识别，依然用旧的标志头封装，那么当用户使用新域名时，就会出现问题。

这里，也许客户会问，如果继续使用非官方SDK（不想更改已有代码），依然使用旧域名，是否可以？

答案是可以，但是不推荐。因为目前控制台已经无法查看到旧域名，如果用户又不知道旧域名是什么，那么需要用户在官网寻求技术帮助。

这里建议用户使用官方SDK，一方面能够得到有保证的支持，另一方面官方SDK提供了BatchSendMessage、BatchReceiveMessage等批量操作，可以带来更高的消息处理效率，而这些功能是没有包含在非官方SDK包里的。

如果用户使用子账号访问，那么需要注意主账号的RAM配置是否正确。

例如：

主账号RAM配置中授权的资源是

```
"Resource": [
  "acs:mns:*:*:/queues",
  "acs:mns:*:*:/queues/busi-test",
  "acs:mns:*:*:/queues/busi-test/*"
```

但实际访问的子账号队列是/queues/busi-msn/messages。那么也会由于不匹配，导致403 AccessDenied。

如问题还未解决，请联系[售后技术支持](#)。

4.7 MNS签名报错403，SignatureDoesNotMatch

您调用MNS HTTP API发送请求时，需要在Authorization header里带上签名值，当签名值和服务器端计算的不一致时，请求会被拒绝，服务器返回403 SignatureDoesNotMatch。本文介绍报错后如何进行检查。

当遇到这种问题，请检查以下几项：

1. 请求URL是否正确。

以ReceiveMessage (curl) 为例：

```
curl -X GET http://44404.mns.cn-beijing.aliyuncs.com/queues/queue1/messages?waitseconds=10 -H "Authorization:MNS ijlkhhklshiojo....." -H "Date:Mon, 09 Nov 2015 06:24:41 GMT" -H "x-mns-version:2015-06-06"
```

- 新协议 (x-mns-version:2015-06-06, Authorization:MNS xxxxxxxx) URL为：

`http://$AccountID.mns.cn-beijing.aliyuncs.com/queues/$queueName/messages`

- 旧协议 (x-mqs-version:2014-07-08, Authorization:MQS xxxxxxxx) URL为：

`http://$AccountID.mns.cn-beijing.aliyuncs.com/$queueName/messages`

新旧协议更多差异，请参见[API参考](#)。

队列地址中的Account ID即云账户的账号ID。



2. AccessKeyId和AccessKeySecret是否正确。

3. 加密信息源是否正确。

下文给出常见的错误信息源例子。

- 错误一：

```
GET
\n
text/xml;charset=utf-8
Date:Thu, 21 May 2015 03:50:03 GMT
x-mqs-version:2014-07-08

/queue1/messages?waitseconds=10
```

- 正确一：

```
GET
\n
```

```
text/xml;charset=utf-8
Thu, 21 May 2015 03:50:03 GMT
x-mqs-version:2014-07-08

/queue1/messages?waitseconds=10
```

解决方法：去掉 “Date:” 。

- 错误二：

```
GET

\n

text/xml;charset=utf-8
Thu, 21 May 2015 03:50:03 GMT
x-mns-version:2015-06-06

/queue1/messages?waitseconds=10
```

- 正确二：

```
GET

\n

text/xml;charset=utf-8
Thu, 21 May 2015 03:50:03 GMT
x-mns-version:2015-06-06

/queues/queue1/messages?waitseconds=10
```

解决方法：如果使用新协议，请使用新协议支持的请求格式。

- 错误三：

```
GET

\n

text/xml;charset=utf-8
Thu, 21 May 2015 03:58:23 GMT
\n

/queues/queue1/messages?waitseconds=10
```

- 正确三：

```
GET

\n

text/xml;charset=utf-8
Thu, 21 May 2015 03:58:23 GMT

/queues/queue1/messages?waitseconds=10
```

解决方法：如果没有以x-mns-开头的head，直接置为空，没有\n。

Signature计算方法，请参见[#unique_51](#)。

官网提供了一个Demo，方便用户验证自己的签名过程是否正确，请参见[MNS常见问题](#)中的问题为什么我的签名总是没算对？

如果问题还未能解决，请联系[售后技术支持](#)。

4.8 MNS消费端代码如何实现消息监听

本文主要介绍MNS的消费端向队列拉取消息实现消息监听的方式。

MNS的消息消费模式为拉（pull），消费端需要向队列循环拉取消息。

一般有两种拉取方式，这里以Java为例：

- 第一种方式，利用定时器，如quartz、Timer、ScheduledExecutorService，每隔一段时间拉取一次消息。
- 第二种方式，使用while循环：

```
while(true){
    try {
        Message msg = queue.popMessage();
        if(msg == null){
            \\PollingWaitSeconds expire
            \\And you could do some work here or do nothing according to your business

        }else{
            String messageBody = msg.getMessageBodyAsString();
            String receiptHandle = msg.getReceiptHandle();
            queue.deleteMessage(receiptHandle);
        }
    } catch (ServiceException ex) {
        ex.printStackTrace();
    } catch (ClientException e){
        e.printStackTrace();
    }finally {
        if (client != null) {
            client.close();
        }
    }
}
```

捕获程序异常后，需要进行相应处理，以增加应用程序健壮性。

这边推荐使用PollingWaitSeconds优化性能。

PollingWaitSeconds：当队列消息量为空时，针对该队列的ReceiveMessage（RESTful API）、popMessage（SDK）等请求最长的等待时间（0-30秒，默认0秒）。

如果超时，SDK返回null（如上代码），RESTful API返回404。

如果问题还未能解决，请联系[售后技术支持](#)。