ALIBABA CLOUD

阿里云

短信服务

SDK参考（新版）

文档版本：20200924

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**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `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

1.短信服务SDK简介

短信服务新版SDK源码已经托管至开源平台Github及主流依赖仓库，推荐使用各语言主流的依赖管理工具安装，或通过GitHub clone的方式使用SDK。所有SDK均只依赖SDK核心库，您可以直接使用SDK核心库，使用CommonRequest方式进行调用，Demo代码可通过OpenAPI Explorer生成。

目前，短信服务提供以下编程语言的SDK。

> 说明
> - 通过JAVA SDK拉取消息回执，请使用JAVA MNS SDK。
> - 通过Node.js SDK拉取消息回执，请使用原Node.js MNS SDK。

SDK	安装说明	DEMO
Java SDK	安装Java SDK	Java SDK DEMO
.NET SDK	安装.NET SDK	.NET SDK DEMO
PHP SDK	安装PHP SDK	PHP SDK DEMO
Python SDK	安装Python SDK	Python SDK DEMO
Node.js SDK	安装Node.js SDK	Node.js SDK DEMO
Go SDK	安装Go SDK	Go SDK DEMO

短信服务SDK升级说明

2019年1月22日，短信服务发布了新版SDK，此日期后接入SDK的用户使用的均为新版SDK；2019年1月22日前接入的SDK的用户使用的是旧版SDK。相较于旧版SDK，新版SDK有以下优势：

- 融入阿里云OpenAPI技术体系，改善用户体验。通过新版短信服务SDK，用户可以使用阿里云资源如API Explorer、CloudShell等工具体验API。
- 阿里云统一维护，降低故障风险。新版短信服务SDK由阿里云SDK统一维护，同时支持自动更新及故障预警，包括Github通知和Deprecated提示等。

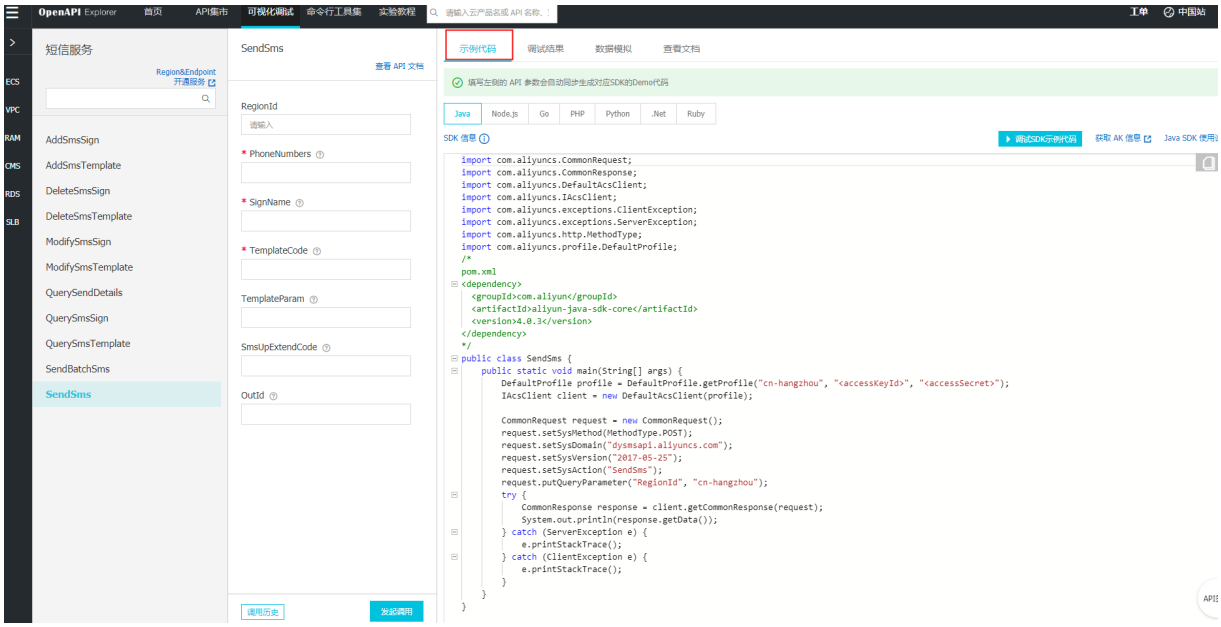
新版短信服务SDK提供了更稳定、完善的开发环境，建议您尽快升级SDK。

2.使用示例（Demo）

短信服务支持通过OpenAPI Explorer快速调用短信服务API。同时在OpenAPI Explorer中提供了多种语言的Demo代码。在左侧填写API 参数后，会在**示例代码**页签中自动同步生成对应SDK的Demo代码。

查看Demo

1. 使用阿里云账号登录OpenAPI Explorer。
2. 在**全部产品**中找到**短信服务**。
3. 单击选择需要查看Demo的API接口。
4. 在左侧参数列中指定参数值，右侧**示例代码**页签中会自动同步生成对应SDK的Demo代码。



3.安装.NET SDK

您可以通过直接添加Nuget 程序包依赖或下载阿里云.NET SDK开发工具包的方式安装阿里云.NET SDK。

> 说明 无论您使用哪种安装方式安装.NET SDK，都必须安装阿里云.NET SDK核心库。

前提条件

在安装和使用阿里云SDK前，确保您已经注册阿里云账号并生成访问密钥（AccessKey）。详情参考创建AccessKey。

安装方式

您可以通过以下两种方式安装.NET SDK。

- 使用依赖包工具安装（推荐）
- 自行下载安装

使用依赖包工具安装（推荐）

您可以通过 NuGet程序包管理器来安装，在解决方案资源管理器面板中右击您的项目选择管理 NuGet程序包菜单，在打开的 NuGet管理面板中点击浏览选项卡输入aliyun-net-sdk-core，选择并点击安装即可。

或通过.NET CLI工具来安装核心库：

```
dotnet add package aliyun-net-sdk-core
```

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

自行下载安装

您可以使用 `git clone` 或其它手段下载aliyun-net-sdk-core并自行添加解决方案。

aliyun-net-sdk-core GitHub地址：aliyun-net-sdk-core。

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

.NET SDK核心库 GitHub地址

.NET SDK核心库

4.安装Python SDK

您可以通过直接添加依赖包或下载阿里云SDK的方式安装阿里云Python SDK。

> 说明 无论您使用哪种安装方式安装Python SDK，都必须安装阿里云Python SDK核心库。

前提条件

在安装和使用阿里云SDK前，确保您已经注册阿里云账号并生成访问密钥（AccessKey）。详情参考创建AccessKey。

安装方式

您可以通过以下两种方式安装Python SDK。

- 使用依赖包工具安装（推荐）
- 自行下载安装

使用依赖包工具安装（推荐）

执行以下命令，安装阿里云SDK核心库。

- 如果您使用Python 2.x，执行以下命令，安装阿里云SDK核心库：

```
pip install aliyun-python-sdk-core
```

- 如果您使用Python 3.x，执行以下命令，安装阿里云SDK核心库：

```
pip install aliyun-python-sdk-core-v3
```

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

自行下载安装

您可以使用git clone或其它手段下载aliyun-python-sdk-core并自行添加解决方案。

aliyun-python-sdk-core GitHub地址：aliyun-python-sdk-core。

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

Python SDK核心库 GitHub地址

Python SDK核心库

5.安装Java SDK

您可以通过直接添加Maven依赖或下载阿里云Java SDK的开发工具包的方式安装阿里云Java SDK。

前提条件

在安装和使用阿里云Java SDK前，确保您已经：

- 安装Java环境。阿里云Java SDK要求使用JDK1.6或更高版本。

  在Java运行环境配置好的情况下，打开windows的命令行，执行*Java -version*命令，可以检查版本信息。

- 已经注册阿里云账号并生成访问密钥（AccessKey）。详细步骤请参考AccessKey。
- 已经安装Java SDK核心库。

Java SDK安装方式

您可以通过以下两种方式安装Java SDK。

- 导入Maven依赖，适用于使用Maven管理的Java项目。
- 在集成开发环境中导入JAR文件，适用于使用Eclipse或IntelliJ作为集成开发环境的项目。

导入Maven依赖

通过在*pom.xml*文件中添加Maven依赖安装Java 阿里云SDK。您可以在Maven库中查看各云产品的Maven依赖信息。

添加以下依赖安装阿里云Java SDK核心库。

```
<dependency>
 <groupId>com.aliyun</groupId>
 <artifactId>aliyun-java-sdk-core</artifactId>
 <version>4.5.3</version>
</dependency>
```

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

在集成开发环境中导入JAR文件

通过导入aliyu-java-sdk-core JAR文件的方式安装阿里云Java SDK。

- Eclipse

  使用Eclipse完成以下操作，在Eclipse的项目中安装阿里云Java SDK：

  i. 将下载的*aliyun-java-sdk-core.jar*文件复制到您的项目文件夹中。

  ii. 在Eclipse中打开您的项目，右键单击该项目，单击Properties。

iii. 在弹出的对话框中，单击Java Build Path > Libraries > Add JARs添加下载的JAR文件。

iv. 单击Apply and Close。

- IntelliJ

  使用IntelliJ完成以下操作，在IntelliJ的项目中安装阿里云Java SDK。

  i. 将下载的*aliyun-java-sdk-core.jar*文件复制到您的项目文件夹中。

  ii. 在IntelliJ中打开您的项目，在菜单栏中单击File > Project Structure。

  iii. 单击Apply，然后单击OK。

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

6.安装Go SDK

可以通过直接安装依赖包的方式安装阿里云Go SDK。

> 说明 无论通过何种方式安装Go SDK，都必须安装阿里云Go SDK核心库。

前提条件

在安装和使用阿里云SDK前，确保您已经注册阿里云账号并生成访问密钥（AccessKey）。详情参考创建AccessKey。

安装方式

您可以通过以下两种方式安装Go SDK。

- 使用依赖包工具安装（推荐）
- 自行下载安装

使用Glide安装GO SDK（推荐）

执行以下命令，安装阿里云Go SDK：

```
glide get github.com/aliyun/alibaba-cloud-sdk-go
```

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

使用Govendor安装

执行以下命令，安装阿里云Go SDK：

```
go get -u github.com/aliyun/alibaba-cloud-sdk-go/sdk
```

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中。

Go SDK GitHub地址

Go SDK核心库

7.安装Node.js SDK

您可以通过直接添加阿里云SDK依赖包的方式安装阿里云Node.js SDK。

> 说明 无论您使用哪种安装方式安装Node.js SDK，都必须安装阿里云Node.js SDK核心库。

前提条件

在安装和使用阿里云Node.js SDK前，确保您已经：

- 安装Node.js环境。请确认Node.js版本为8.0及以上。
- 已经注册阿里云账号并生成访问密钥（AccessKey）。详细步骤请参考创建AccessKey。

安装步骤

请参考以下步骤，使用依赖包工具安装Node.js SDK。

执行以下命令，安装阿里云SDK核心库。

```
npm install @alicloud/pop-core -S
```

您也可以使用cnpm、yarn等包管理工具来安装@alicloud/pop-core。

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo并应用在您的项目中，如需了解更多，请参考GitHub仓库说明。

Node.js SDK GitHub地址

Node.js SDK核心库

8.安装PHP SDK

您可以通过直接添加阿里云SDK依赖包的方式安装阿里云PHP SDK。

前提条件

在安装和使用阿里云PHP SDK前，确保您已经：

- 注册阿里云账号并生成访问密钥（AccessKey）。详细步骤请参考AccessKey。
- 使用阿里云PHP SDK调用短信服务的API前，确保您已经在阿里云控制台开通了短信服务。
- 安装PHP环境。阿里云PHP SDK适用于PHP 5.5.0或更高版本。 您可以通过 `php -v` 命令查看当前使用的PHP版本号。
- 安装cURL，并使用TLS后端编译cURL 7.16.2+。

安装步骤

请参考以下步骤，使用Composer安装依赖。

如果在您的系统上全局安装Composer，您可以在项目目录中运行以下内容，将 Alibaba Cloud Client for PHP 添加为依赖项。

```
composer require alibabacloud/client
```

通过 Composer 和其他方式安装的详细操作，请查看安装说明。

在安装完成后，您可以使用OpenAPI Explorer来生成相关API的Demo，并应用在您的项目中。如需了解更多，请参考更详细的安装及使用指南。

PHP SDK GitHub地址

PHP SDK核心库

9.升级SDK

新SDK源码已经托管至开源平台Github，可使用GitHub clone的方式使用SDK，也可以使用依赖管理工具安装，Demo代码可通过OpenAPI Explorer生成，所有SDK均只依赖SDK核心库，使用通用的Request及Response来处理接口请求及响应。

JAVA SDK

如何安装JAVA SDK：安装Java SDK。

新版JAVA SDK仅依赖阿里云JAVA SDK核心库，可使用maven安装依赖，使用CommonRequest、CommonResponse来处理接口请求及响应。请使用CommonRequest实例的putQueryParameter方法设置API参数，CommonResponse实例使用getData方法获取API结果的JSON字符串。

> 说明 通过JAVA SDK拉取消息回执，请使用JAVA MNS SDK。

Node.js SDK

如何安装Node.js SDK：安装Node.js SDK。

详情：@alicloud/pop-core。新Node.js SDK仅依赖@alicloud/pop-core。

> 说明 通过Node.js SDK拉取消息回执，请使用原Node.js MNS SDK。

.NET SDK

如何安装.NET SDK：安装.NET SDK。

新版.NET SDK仅依赖阿里云.NET SDK核心库，可使用NuGet安装依赖，使用CommonRequest、CommonResponse来处理接口请求及响应。 请使用CommonRequest实例的putQueryParameter方法设置API参数，CommonResponse实例使用getData方法获取API结果的JSON字符串。

PHP SDK

如何安装PHP SDK：安装PHP SDK。

PHP全新SDK上线，并支持composer安装，请按说明重新设置请求参数。

Python SDK

如何安装Python SDK：安装Python SDK。

python-sdk-core已统一支持Python 2.0/3.0，与原有SDK兼容，建议参考OpenAPI Explorer提供的Demo重新设置请求参数。

Go SDK

如何安装Go SDK：安装Go SDK。

原SDK不支持Golang，使用新SDK请直接参考新Demo即可。

10.升级消息回执API

10.1. 升级说明

升级前，请注意以下事项：

已接入SDK的老用户或Java、Node.JS用户无需升级消息回执API，新用户的消息回执接入请参考现有Demo。

10.2. .NET Demo

依赖.NET语言的阿里云SDK核心库及dybaseapi，其中dybaseapi包用于拉取MNS消息。

- 阿里云SDK核心库
- dybaseapi

Demo如下：

```
using System;
using Aliyun.Acs.Core;
using Aliyun.Acs.Core.Profile;
using Aliyun.Acs.Core.Exceptions;
using Aliyun.Acs.Dybaseapi.Model.V20170525;
using Aliyun.Acs.Dybaseapi.MNS;
using Aliyun.Acs.Dybaseapi.MNS.Model;
using System.Threading;
using System.Collections.Generic;
using System.Text;
using QueryTokenForMnsQueue_MessageTokenDTO = Aliyun.Acs.Dybaseapi.Model.V20170525.QueryTo
kenForMnsQueueResponse.QueryTokenForMnsQueue_MessageTokenDTO;

namespace CommonRpc
{
  class Program
  {
    static void Main(string[] args)
    {
      IClientProfile profile = DefaultProfile.GetProfile("cn-hangzhou", "<AccessKeyId>", "<AccessKeyS
ecret>"); // todo: 补充AK信息

      DefaultProfile.AddEndpoint("cn-hangzhou", "cn-hangzhou", "Dybaseapi", "dybaseapi.aliyuncs.co
m");

      DefaultAcsClient client = new DefaultAcsClient(profile);
```

```
        String queueName = "<QueueName>"; // todo: 补充队列名称
        String messageType = "<MessageType>"; // todo: 补充消息类型

        int maxThread = 2;

        for (int i = 0; i < maxThread; i++)
        {
          TestTask testTask = new TestTask("PullMessageTask-thread-" + i, messageType, queueNam
e, client);
          Thread t = new Thread(new ThreadStart(testTask.Handle));
          //启动线程
          t.Start();
        }
        Console.ReadKey();

        try
        {
          QueryTokenForMnsQueueRequest request = new QueryTokenForMnsQueueRequest
          {
            MessageType = messageType,
            QueueName = queueName
          };

          QueryTokenForMnsQueueResponse response = client.GetAcsResponse(request);
          Console.WriteLine(response.MessageTokenDTO.SecurityToken);
        }
        catch (ServerException ex)
        {
          Console.WriteLine(ex.ToString());
        }
        catch (ClientException ex)
        {
          Console.WriteLine(ex.ToString());
        }
      }
    }

    class TestTask
    {
      object o = new object();
```

```
    const int sleepTime = 50;
    const long bufferTime = 60 * 2; // 过期时间小于2分钟则重新获取，防止服务器时间误差
    const String mnsAccountEndpoint = "https://1943695596114318.mns.cn-hangzhou.aliyuncs.com/";
// 阿里通信消息的endpoint,固定

    public String name { get; private set; }
    public String messageType { get; private set; }
    public String QueueName { get; private set; }
    public int TaskID { get; private set; }
    public IAcsClient AcsClient { get; private set; }

    public TestTask(String name, String messageType, String queueName, IAcsClient acsClient)
    {
      this.name = name;
      this.messageType = messageType;
      this.QueueName = queueName;
      this.AcsClient = acsClient;
    }

    readonly Dictionary<string, QueryTokenForMnsQueue_MessageTokenDTO> tokenMap = new Dicti
onary<string, QueryTokenForMnsQueue_MessageTokenDTO>();
    readonly Dictionary<string, Queue> queueMap = new Dictionary<string, Queue>();

    public QueryTokenForMnsQueue_MessageTokenDTO GetTokenByMessageType(IAcsClient acsClie
nt, String messageType)
    {
      QueryTokenForMnsQueueRequest request = new QueryTokenForMnsQueueRequest
      {
        MessageType = messageType
      };
      QueryTokenForMnsQueueResponse queryTokenForMnsQueueResponse = acsClient.GetAcsResp
onse(request);
      QueryTokenForMnsQueue_MessageTokenDTO token = queryTokenForMnsQueueResponse.Mes
sageTokenDTO;
      return token;
    }

    /// 处理消息
    public void Handle()
    {
      while (true)
```

```
        {
            try
            {
                QueryTokenForMnsQueue_MessageTokenDTO token = null;
                Queue queue = null;
                lock (o)
                {
                    if (tokenMap.ContainsKey(messageType))
                    {
                        token = tokenMap[messageType];
                    }

                    if (queueMap.ContainsKey(QueueName))
                    {
                        queue = queueMap[QueueName];
                    }

                    TimeSpan ts = new TimeSpan(0);

                    if (token != null)
                    {
                        DateTime b = Convert.ToDateTime(token.ExpireTime);
                        DateTime c = Convert.ToDateTime(DateTime.Now);
                        ts = b - c;
                    }

                    if (token == null || ts.TotalSeconds < bufferTime || queue == null)
                    {
                        token = GetTokenByMessageType(AcsClient, messageType);
                        IMNS client = new MNSClient(token.AccessKeyId, token.AccessKeySecret, mnsAccountE
ndpoint, token.SecurityToken);
                        queue = client.GetNativeQueue(QueueName);
                        if (tokenMap.ContainsKey(messageType))
                        {
                            tokenMap.Remove(messageType);
                        }
                        if (queueMap.ContainsKey(QueueName))
                        {
                            queueMap.Remove(QueueName);
                        }
```

```
                tokenMap.Add(messageType, token);
                queueMap.Add(QueueName, queue);
            }
        }

        BatchReceiveMessageResponse batchReceiveMessageResponse = queue.BatchReceiveMes
sage(16);
        List<Message> messages = batchReceiveMessageResponse.Messages;

        for (int i = 0; i <= messages.Count - 1; i++)
        {
            try
            {
                byte[] outputb = Convert.FromBase64String(messages[i].Body);
                string orgStr = Encoding.UTF8.GetString(outputb);
                Console.WriteLine(orgStr);
                // TODO 具体消费逻辑,待客户自己实现.
                // 消费成功的前提下删除消息
                // queue.DeleteMessage(messages[i].ReceiptHandle);
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.ToString());
        }
        Thread.Sleep(sleepTime);
      }
    }
  }
}
```

10.3. PHP Demo

依赖OpenAPI PHP Client包和OpenAPI PHP SDK包。

- OpenAPI PHP Client

- 下载地址：openapi-sdk-php-client
- 安装说明OpenAPI PHP Client 安装说明

- OpenAPI PHP SDK
  - 下载地址：openapi-sdk-php
  - 安装说明OpenAPI PHP SDK 安装说明

Demo如下：

```
<?php
use AlibabaCloud\Client\AlibabaCloud;
use AlibabaCloud\Client\Exception\ClientException;
use AlibabaCloud\Client\Exception\ServerException;
use AlibabaCloud\Dybaseapi\MNS\Requests\BatchReceiveMessage;
use AlibabaCloud\Dybaseapi\MNS\Requests\BatchDeleteMessage;

AlibabaCloud::accessKeyClient('<AccessKeyId>', '<AccessSecret>')
    ->regionId('cn-hangzhou')
    ->asGlobalClient();

$queueName = '<QueueName>'; // 队列名称
$messageType = '<MessageType>'; // 需要接收的消息类型

$response = null;
$token = null;
$i = 0;

do {
    try {
        if (null == $token || strtotime($token['ExpireTime']) - time() > 2 * 60) {
            $response = AlibabaCloud::rpcRequest()
                ->product('Dybaseapi')
                ->version('2017-05-25')
                ->action('QueryTokenForMnsQueue')
                ->method('POST')
                ->host("dybaseapi.aliyuncs.com")
                ->options([
                    'query' => [
                        'MessageType' => $messageType,
                        'QueueName' => $queueName,
                    ],
                ])
```

```
            ->request()
            ->toArray();
        }

        $token = $response['MessageTokenDTO'];

        $mnsClient = new \AlibabaCloud\Dybaseapi\MNS\MnsClient(
            "http://1943695596114318.mns.cn-hangzhou.aliyuncs.com",
            $token['AccessKeyId'],
            $token['AccessKeySecret'],
            $token['SecurityToken']
        );
        $mnsRequest = new BatchReceiveMessage(10, 5);
        $mnsRequest->setQueueName($queueName);
        $mnsResponse = $mnsClient->sendRequest($mnsRequest);

        $receiptHandles = Array();
        foreach ($mnsResponse->Message as $message) {
            // 用户逻辑:
            // $receiptHandles[] = $message->ReceiptHandle; // 加入$receiptHandles数组中的记录将会被删除
            $messageBody = base64_decode($message->MessageBody); // base64解码后的JSON字符串
            print_r($messageBody . "\n");
        }

        if (count($receiptHandles) > 0) {
            $deleteRequest = new BatchDeleteMessage($queueName, $receiptHandles);
            $mnsClient->sendRequest($deleteRequest);
        }
    } catch (ClientException $e) {
        echo $e->getErrorMessage() . PHP_EOL;
    } catch (ServerException $e) {
        if ($e->getCode() == 404) {
            $i++;
        }
        echo $e->getErrorMessage() . PHP_EOL;
    }
} while ($i < 3);
```

10.4. Python Demo

依赖Python语言的阿里云SDK核心库及dybaseapi，其中dybaseapi包用于拉取MNS消息。

- 阿里云SDK核心库（Python）
- dybaseapi（Python）

Demo如下：

```
#!/usr/bin/env python
# coding=utf8

import time
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkcore.client import AcsClient
from aliyunsdkdybaseapi.request.v20170525.QueryTokenForMnsQueueRequest import QueryTokenFor
MnsQueueRequest
from aliyunsdkcore.profile import region_provider
from datetime import datetime
from aliyunsdkdybaseapi.mns.account import Account
from aliyunsdkdybaseapi.mns.queue import *
from aliyunsdkdybaseapi.mns.mns_exception import *

try:
    import json
except ImportError:
    import simplejson as json


# TODO 需要替换成您需要接收的消息类型
message_type = "<MessageType>"
# TODO 需要替换成您的队列名称。在云通信页面开通相应业务消息后，就能在页面上获得对应的queueName
queue_name = "<QueueName>"

# 云通信固定的endpoint地址
endpoint = "http://1943695596114318.mns.cn-hangzhou.aliyuncs.com"

acs_client = AcsClient("<AccessKeyId>", "<AccessKeySecret>", "cn-hangzhou")
region_provider.add_endpoint("Dybaseapi", "dybaseapi.aliyuncs.com", "cn-hangzhou")


# 云通信业务token存在失效时间，需动态更新。
class Token():
    def __init__(self):
        self.token = None
```

```
        self.tmp_access_id = None
        self.tmp_access_key = None
        self.expire_time = None

    def is_refresh(self):
        if self.expire_time is None:
            return 1
        # 失效时间与当前系统时间比较，提前2分钟刷新token
        now = datetime.now()
        expire = datetime.strptime(self.expire_time, "%Y-%m-%d %H:%M:%S")
        if expire <= now or (expire - now).seconds < 120:
            return 1
        return 0

    def refresh(self):
        print("start refresh token...")
        request = QueryTokenForMnsQueueRequest()
        request.set_MessageType(message_type)
        request.set_QueueName(queue_name)
        response = acs_client.do_action_with_exception(request)
        # print response
        if response is None:
            raise ServerException("GET_TOKEN_FAIL", "获取token时无响应")

        response_body = json.loads(response)

        if response_body.get("Code") != "OK":
            raise ServerException("GET_TOKEN_FAIL", "获取token失败")

        sts_token = response_body.get("MessageTokenDTO")
        self.tmp_access_key = sts_token.get("AccessKeySecret")
        self.tmp_access_id = sts_token.get("AccessKeyId")
        self.expire_time = sts_token.get("ExpireTime")
        self.token = sts_token.get("SecurityToken")

        print("finish refresh token...")


# 初始化 token, my_account, my_queue
token, my_account, my_queue = Token(), None, None
```

```
# 循环读取删除消息直到队列空
# receive message请求使用long polling方式，通过wait_seconds指定长轮询时间为3秒

## long polling 解析:
### 当队列中有消息时，请求立即返回;
### 当队列中没有消息时，请求在MNS服务器端挂3秒钟，在这期间，有消息写入队列，请求会立即返回消息，3秒后，请求返回队列没有消息;

wait_seconds = 3
print("%sReceive And Delete Message From Queue%s\nQueueName:%s\nWaitSeconds:%s\n" % (
    10 * "=", 10 * "=", queue_name, wait_seconds))

while True:
    receipt_handles = []
    # 读取消息
    try:
        # token过期是否需要刷新
        if token.is_refresh() == 1:
            # 刷新token
            token.refresh()
            if my_account:
                my_account.mns_client.close_connection()
                my_account = None

        if not my_account:
            my_account = Account(endpoint, token.tmp_access_id, token.tmp_access_key, token.token)
            my_queue = my_account.get_queue(queue_name)

        # 接收消息
        recv_msgs = my_queue.batch_receive_message(10, wait_seconds)

        for recv_msg in recv_msgs:
            # TODO 业务处理

            # receipt_handles.append(recv_msg.receipt_handle)
            print("Receive Message Succeed! ReceiptHandle:%s MessageBody:%s MessageID:%s" % (
                recv_msg.receipt_handle, recv_msg.message_body, recv_msg.message_id))

    except MNSExceptionBase as e:
        if e.type == "QueueNotExist":
```

```
            print("Queue not exist, please create queue before receive message.")
            break
        elif e.type == "MessageNotExist":
            print("Queue is empty! sleep 10s")
            time.sleep(10)
            continue
        print("Receive Message Fail! Exception:%s\n" % e)
        break

    # 删除消息
    try:
        if len(receipt_handles) > 0:
            # my_queue.delete_message(receipt_handles)
            print("Delete Message Succeed!  ReceiptHandles:%s" % receipt_handles)
    except MNSExceptionBase as e:
        print("Delete Message Fail! Exception:%s\n" % e)
```

10.5. Go Demo

SDK：alibaba-cloud-sdk-go

Demo如下：

```
package main

import (
 "encoding/base64"
 "fmt"
 "github.com/aliyun/alibaba-cloud-sdk-go/sdk/endpoints"
 "github.com/aliyun/alibaba-cloud-sdk-go/services/dybaseapi"
 "github.com/aliyun/alibaba-cloud-sdk-go/services/dybaseapi/mns"
 "time"
)

const (
 mnsDomain = "1943695596114318.mns.cn-hangzhou.aliyuncs.com"
)

func main() {
 endpoints.AddEndpointMapping("cn-hangzhou", "Dybaseapi", "dybaseapi.aliyuncs.com")

 // 创建client实例
```

```
client, err := dybaseapi.NewClientWithAccessKey(
 "cn-hangzhou",          // 您的可用区ID
 "<AccessKeyId>",        // 您的Access Key ID
 "<AccessKeySecret>")    // 您的Access Key Secret
if err != nil {
 // 异常处理
 panic(err)
}

queueName := "<QueueName>"
messageType := "<MessageType>"

var token *dybaseapi.MessageTokenDTO

for {
 if token == nil || token.ExpireTime - time.Now().Unix() > 2 * 60 {
  // 创建API请求并设置参数
  request := dybaseapi.CreateQueryTokenForMnsQueueRequest()
  request.MessageType = messageType
  request.QueueName = queueName
  // 发起请求并处理异常
  response, err := client.QueryTokenForMnsQueue(request)
  if err != nil {
   // 异常处理
   panic(err)
  }

  token = &response.MessageTokenDTO
 }

 mnsClient, err := mns.NewClientWithStsToken(
  "cn-hangzhou",
  token.AccessKeyId,
  token.AccessKeySecret,
  token.SecurityToken,
 )

 if err != nil {
  panic(err)
 }
```

```
 mnsRequest := mns.CreateBatchReceiveMessageRequest()
 mnsRequest.Domain = mnsDomain
 mnsRequest.QueueName = queueName
 mnsRequest.NumOfMessages = "10"
 mnsRequest.WaitSeconds = "5"

 mnsResponse, err := mnsClient.BatchReceiveMessage(mnsRequest)
 if err != nil {
  panic(err)
 }
 // fmt.Println(mnsResponse)

 receiptHandles := make([]string, len(mnsResponse.Message))
 for i, message := range mnsResponse.Message {
  messageBody, decodeErr := base64.StdEncoding.DecodeString(message.MessageBody)
  if decodeErr != nil {
   panic(decodeErr)
  }
  fmt.Println(string(messageBody))
  receiptHandles[i] = message.ReceiptHandle
 }
 if len(receiptHandles) > 0 {
  mnsDeleteRequest := mns.CreateBatchDeleteMessageRequest()
  mnsDeleteRequest.Domain = mnsDomain
  mnsDeleteRequest.QueueName = queueName
  mnsDeleteRequest.SetReceiptHandles(receiptHandles)
  //_, err = mnsClient.BatchDeleteMessage(mnsDeleteRequest) // 取消注释将删除队列中的消息
  if err != nil {
   panic(err)
  }
 }
}
```