

阿里云 语音服务

SDK参考

文档版本：20200521

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 <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 Instance_ID</code>
[]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

法律声明	I
通用约定	I
1 SDK简介	1
2 使用OpenAPI Explorer	2
3 升级SDK	4
4 安装SDK	6
4.1 Java SDK.....	6
4.2 Python SDK.....	6
4.3 Node.js SDK.....	7
4.4 Go SDK.....	8
4.5 PHP SDK.....	8
4.6 .NET SDK.....	9
5 Java示例	10
5.1 调用SDK.....	10
5.2 语音通知.....	12
5.3 语音IVR.....	14
5.4 点击呼叫.....	15
5.5 取消点击呼叫.....	17
5.6 智能外呼.....	18
5.7 查询通话记录详情.....	19
6 Python示例	21
6.1 调用SDK.....	21
6.2 语音通知.....	23
6.3 语音IVR.....	24
6.4 点击呼叫.....	25
6.5 取消点击呼叫.....	26
6.6 智能外呼.....	27
6.7 查询通话记录详情.....	28
7 消息回执	29
7.1 升级说明.....	29
7.2 .NET Demo.....	29
7.3 PHP Demo.....	32
7.4 Python Demo.....	34
7.5 Go Demo.....	36

1 SDK简介

阿里云语音服务为您提供多种版本SDK，以访问密钥（AccessKey）识别调用者身份，提供自动签名等功能，方便您通过API实现不同功能。

SDK列表

目前，语音服务的最新SDK版本号为2017-05-25，并提供以下编程语言的SDK，建议您直接通过阿里云OpenAPI Explorer工具进行在线调用和调试。更多SDK请访问[阿里云开放平台](#)。



说明：

通过Java SDK拉取消息回执，请使用[MNS SDK](#)。

Alibaba Cloud SDK	OpenAPI Explorer
Alibaba Cloud SDK for Java	Java
Alibaba Cloud SDK for Node.js	Node.js
Alibaba Cloud SDK for Go	Go
Alibaba Cloud SDK for PHP	PHP
Alibaba Cloud SDK for Python	Python
Alibaba Cloud SDK for .NET	.NET

相关链接

- 如何获取AccessKey请参见[#unique_4](#)。
- 更多语言版本的SDK，请前往[第三方SDK服务](#)。
- 语音服务的接入地址、请求签名等信息请参见[API概述](#)。

2 使用OpenAPI Explorer

阿里云语音服务为您提供OpenAPI Explorer在线调用工具，帮助您快速调用SDK进行调用和调试。
本章节以调用SingleCallByTts向指定号码发起语音验证码为您示例。

前提条件

您已经完成注册阿里云账号，并完成实名认证，具体操作请参见[#unique_7](#)。

您需要开通语音服务并完成购买号码等操作，详情请参见[#unique_8](#)。

操作步骤

1. 登录OpenAPI Explorer。
2. 在搜索输入框输入云通信语音服务。



3. 在云通信语音服务页面，输入SingleCallByTts快速定位到该接口具体调用页面。
4. 根据API文档说明，填写各参数取值，右侧示例代码会自动生成对应SDK的Demo代码。

本章节调用接口参数说明，详情请参见[#unique_9](#)

5. 单击发起调用。

- 正常返回示例。

```
{
  "Message": "OK",
  "RequestId": "2886A9EB-44B6-43C6-9E2F-031ECCCE4CEB",
  "Code": "OK",
  "CallId": "118697418557^1054"
}
```

- 异常返回示例。

```
{
  "Message": "文本转语音模板不合法",
  "RequestId": "AE3D4BAE-F95C-4C0B-9C1E-DDF0F49C5645",
  "Code": "isv.TEMPLATE_ILLEGAL"
}
```

3 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: [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: [Node.js SDK](#)。

详情: [@alicloud/pop-core](#)。新Node.js SDK仅依赖[@alicloud/pop-core](#)。

.NET SDK

如何安装.NET SDK: [.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: [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: [Pyhton 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: [Go SDK](#)。

原SDK不支持Golang, 使用新SDK请直接参考新Demo即可。

4 安装SDK

4.1 Java SDK

本章节为您介绍阿里云语音服务Java SDK的安装方式。

前提条件

安装Java SDK之前，请您完成操作：本机已经安装Java环境。



说明：

本章节以开发工具IDEA为您示例。

Maven加载

1. 在**IDEA工程 (Project)** 界面，新建一个Maven工程。
2. 在pom.xml文件中，添加如下依赖。

```
<!--添加阿里云core依赖-->
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.4.4</version>
</dependency>
```



说明：

本文SDK版本仅提供参考，具体版本请您以实际为准。

后续步骤

完成安装，您可以开始使用SDK：

- 了解调用语音服务API，请您参见[#unique_5](#)。
- 了解全部语音服务API，请您参见[#unique_19](#)。

4.2 Python SDK

本章节为您介绍阿里云语音服务Python SDK的安装方式。

前提条件

安装Python SDK之前，请您完成操作：本机已经安装Python环境。

pip安装

您需要本地已安装pip环境。

1. 打开CMD命令窗口。
2. 运行命令。
 - 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
```

后续步骤

完成安装, 您可以开始使用SDK:

- 了解调用语音服务API, 请您参见[#unique_5](#)。
- 了解全部语音服务API, 请您参见[#unique_19](#)。

4.3 Node.js SDK

本章节为您介绍阿里云语音服务Node.js SDK的两种安装方式。

前提条件

安装Node.js SDK之前, 请您完成以下操作:

- 本机已经安装Node.js环境。
- 已经下载并配置Node.js开发工具, 例如: Pycharm。

命令安装

1. 打开CMD命令窗口。
2. 运行命令。

```
$ npm install @alicloud/pop-core --save
```

手动加载Jar包

您需要手动下载aliyun-Node.js-sdk-core包。

1. 下载[SDK](#)。
2. 将本地包加载到工程 (Project) 目录。

后续步骤

完成安装，您可以开始使用SDK：

- 了解调用语音服务API，请您参见[#unique_5](#)。
- 了解全部语音服务API，请您参见[#unique_19](#)。

4.4 Go SDK

本章节为您介绍阿里云语音服务Go SDK的安装方式。

前提条件

安装Go SDK之前，请您完成以下操作：本机已经安装Go环境。

(推荐) Glide方式

使用以下命令安装。

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

Govendor方式

使用以下命令安装。

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

后续步骤

完成安装，您可以开始使用SDK：

- 了解调用语音服务API，请您参见[#unique_5](#)。
- 了解全部语音服务API，请您参见[#unique_19](#)。

4.5 PHP SDK

本章节为您介绍阿里云语音服务PHP SDK的安装方式。

前提条件

安装PHP SDK之前，请您完成操作：本机已经安装[PHP环境](#)。

Composer方式

您需要完成安装[Composer](#)。

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
```

后续步骤

完成安装，您可以开始使用SDK：

- 了解调用语音服务API，请您参见[#unique_5](#)。
- 了解全部语音服务API，请您参见[#unique_19](#)。

4.6 .NET SDK

本章节为您介绍阿里云语音服务.NET SDK的安装方式。

前提条件

安装.NET SDK之前，请您完成操作：本机已经安装.NET环境。

阿里云CLI工具安装

您需要先完成安装CLI工具，请参见[阿里云CLI工具简介](#)。

执行以下命令：

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

NuGet程序包管理器

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管理面板中单击浏览选项卡输入aliyun-net-sdk-core，选择并单击安装即可。

后续步骤

完成安装，您可以开始使用SDK：

- 了解调用语音服务API，请您参见[#unique_5](#)。
- 了解全部语音服务API，请您参见[#unique_19](#)。

5 Java示例

5.1 调用SDK

本章节为您介绍调用阿里云语音服务SDK的具体步骤，您可以参考本文档，调用其他API实现对应功能。

前提条件

在使用语音服务SDK之前，您需要：

- 开通语音服务SDK并购买语音号码，具体操作请参见[#unique_8](#)。
- 获取Accesskey信息，具体操作请参见[#unique_4](#)。
- 完成安装SDK，具体操作请参见[#unique_22](#)。



说明：

本章节以调用SingleCallByTts实现向指定号码发送语音验证码为您示例，API详情请参见[#unique_9](#)。

操作步骤

1. 创建profile和client实例，并传入AccessKey信息。

```
DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
IAcsClient client = new DefaultAcsClient(profile);
```



说明：

RegionId默认选择cn-hangzhou。

2. 创建request实例，并设置请求参数。

```
CommonRequest request = new CommonRequest();
//请求方法分为POST和GET，建议您选择POST方式
request.setMethod(MethodType.POST);
//Domain参数的默认值为dyvmsapi.aliyuncs.com
request.setDomain("dyvmsapi.aliyuncs.com");
//当前语音服务SDK默认版本号为2017-05-25
request.setVersion("2017-05-25");
//被叫显示号码信息
request.putQueryParameter("CalledShowNumber", "1862xxxx");
//被叫号码
request.putQueryParameter("CalledNumber", "02131xxxx");
//文本转语音（TTS）模板ID
request.putQueryParameter("TtsCode", "TTS_177553");
```

```
...
```

3. 创建response实例，并传入请求参数。

```
CommonResponse response = client.getCommonResponse(request);
```

4. 打印输出信息。

```
System.out.println(JSON.toJSONString(response));
```

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
/*
pom.xml
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.0.3</version>
</dependency>
*/
public class SingleCallByTts {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<
accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        CommonRequest request = new CommonRequest();
        request.setMethod(MethodType.POST);
        request.setDomain("dyvmsapi.aliyuncs.com");
        request.setVersion("2017-05-25");
        request.setAction("SingleCallByTts");
        request.putQueryParameter("RegionId", "cn-hangzhou");
        request.putQueryParameter("CalledShowNumber", "1862xxxx");
        request.putQueryParameter("CalledNumber", "0213xxxx");
        request.putQueryParameter("TtsCode", "TTS_177553");
        request.putQueryParameter("PlayTimes", "3");
        request.putQueryParameter("Volume", "60");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

返回示例如下所示。

- 正常返回示例。

```
    {
      "Message": "OK",
      "RequestId": "2886A9EB-44B6-43C6-9E2F-031ECCCE4CEB",
      "Code": "OK",
      "CallId": "118697418557^1054"
    }
  }
```

- 异常返回示例。

```
    {
      "Message": "文本转语音模板不合法",
      "RequestId": "AE3D4BAE-F95C-4C0B-9C1E-DDF0F49C5645",
      "Code": "isv.TEMPLATE_ILLEGAL"
    }
  }
```

5.2 语音通知

本章节为您介绍阿里云语音服务语音通知的功能介绍和实现方法，您可以通过调用SingleCallByVoice实现发起语音文件类型的语音通知、调用SingleCallByTts实现发起文本转语音类型的语音通知。

功能介绍

语音通知可以很好的解决通知到达率延迟、关注度不高、阅读体验差等问题，并且您可以通过电话获得更全面的信息通知。您只需要提前设置通知模版内容，调用API就可以给用户推送语音通知，并可以设置播放音量、播放次数等。

实现方法

在实现调用API实现语音通知功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Java SDK](#)。

本章节调用接口参数详情，请参见[#unique_24](#)、[#unique_9](#)。

```
import com.alibaba.fastjson.JSON;
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
```



```
public class yuyinTest {
    private static String SingleCallByVoice (IAcsClient client,String Domain,String Version ,
    MethodType Method) throws ClientException{
        CommonRequest request = new CommonRequest();
        request.setDomain(Domain);
        request.setVersion(Version);
        request.setMethod(Method);
        request.setAction("SingleCallByVoice");
        request.putQueryParameter("CalledShowNumber", "18850xxxx");
        request.putQueryParameter("CalledNumber", "15750xxxx");
        request.putQueryParameter("VoiceCode", "e271f3f2-e155-4366-a9f4-0fe55765b3ec.
wav");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println("发起语音文件通知返回");
            System.out.println(JSON.toJSONString(response.getData()));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
        return null;
    }
    private static String SingleCallByTts (IAcsClient client,String Domain,String Version ,
    MethodType Method) throws ClientException{
        CommonRequest request = new CommonRequest();
        request.setDomain(Domain);
        request.setVersion(Version);
        request.setMethod(Method);
        request.setAction("SingleCallByTts");
        request.putQueryParameter("CalledShowNumber", "18850xxxx");
        request.putQueryParameter("CalledNumber", "15750xxxx");
        request.putQueryParameter("TtsCode", "TTS_130555018");
        request.putQueryParameter("TtsParam", "{ \"name\": \"测试\", \"time\": \"2019年\"}");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println("文本转语音类型返回");
            System.out.println(JSON.toJSONString(response.getData()));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
        return null;
    }
}

public static void main(String []args) throws Exception {
    //创建profile对象, 并传入AccessKey信息, RegionId默认cn-hangzhou
    DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "LTAIxxxx", "
Q3STpxxxx");
    IAcsClient client = new DefaultAcsClient(profile);
    //设置请求接口域名、版本号、请求方式
    final String Domain = "dyvmsapi.aliyuncs.com";
    final String Version = "2017-05-25";
    final MethodType Method = MethodType.POST;
    String SingleCallByVoice = SingleCallByVoice(client,Domain,Version,Method);
    String SingleCallByTts = SingleCallByTts(client,Domain,Version,Method);
}
}
```

正常返回结果如下所示:

发起语音文件通知返回

```
{
  "Message": "OK",
  "RequestId": "4ABF7FB6-449C-4624-8CC6-9AAE368453AA",
  "Code": "OK",
  "CallId": "118720962490^10551812xx"
}
文本转语音类型返回
{
  "Message": "OK",
  "RequestId": "762104B0-D58A-4D04-8486-7C5171D0D756",
  "Code": "OK",
  "CallId": "118720797955^10551795xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

5.3 语音IVR

本章节为您介绍阿里云语音服务语音IVR的功能介绍和实现方法，您可以通过调用IvrCall实现交互式语音应答。

功能介绍

语音IVR是指通过调用语音呼叫的API，从运营商网络向指定号码发起一通呼叫，呼叫被接起后，播放一段指定音频，用户根据音频引导，通过手机按键信息返回意图，语音平台通过消息回执返回按键信息给企业业务系统。

实现方法

在实现调用API实现语音IVR功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Java SDK](#)。

本章节调用接口详情，请参见[#unique_27](#)。

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class IvrCall {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        CommonRequest request = new CommonRequest();
        request.setMethod(MethodType.POST);
```

```
request.setDomain("dyvmsapi.aliyuncs.com");
request.setVersion("2017-05-25");
request.setAction("IvrCall");
request.putQueryParameter("RegionId", "cn-hangzhou");
request.putQueryParameter("CalledShowNumber", "05715621xx");
request.putQueryParameter("CalledNumber", "1500000xx");
request.putQueryParameter("StartCode", "TTS_71390000");
request.putQueryParameter("StartTtsParams", "{ \"product\": \"aliyun\", \"code\": \"123\" }");
request.putQueryParameter("MenuKeyMap.1.Key", "1");
request.putQueryParameter("MenuKeyMap.1.Code", "9a9d7222-670f-40b0-a3af.wav");
request.putQueryParameter("MenuKeyMap.2.Key", "2");
request.putQueryParameter("MenuKeyMap.2.Code", "TTS_71390000");
request.putQueryParameter("MenuKeyMap.2.TtsParams", "{ \"product\": \"aliyun\", \"code\": \"123\" }");
request.putQueryParameter("PlayTimes", "3");
request.putQueryParameter("ByeTtsParams", "{ \"product\": \"aliyun\", \"code\": \"123\" }");
request.putQueryParameter("Timeout", "3000");
try {
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "4ABF7FB6-449C-4624-8CC6-9AAE368453AA",
  "Code": "OK",
  "CallId": "118720962490^10551812xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

5.4 点击呼叫

本章节为您介绍如何调用ClickToDial实现阿里云语音通知点击呼叫功能。

功能介绍

点击呼叫是指通过调用语音服务接口，通过语音服务分配的号码分别向主叫、被叫发起呼叫，双方接通后建立起正常通话，通话双方显示的号码均为语音服务平台号码。常用于企业办公电话等，例如钉钉办公电话。通过点击呼叫接口，可隐藏通话双方真实号码，同时平台可留存双方通话记录。

实现方法

实现点击呼叫功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Java SDK](#)。

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
import com.alibaba.fastjson.JSON;

public class ClickToDial {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        CommonRequest request = new CommonRequest();
        request.setMethod(MethodType.POST);
        request.setDomain("dyvmsapi.aliyuncs.com");
        request.setVersion("2017-05-25");
        request.setAction("ClickToDial");
        request.putQueryParameter("CallerShowNumber", "123xxxx");
        request.putQueryParameter("CallerNumber", "1575xxxx");
        request.putQueryParameter("CalledShowNumber", "12xxxx");
        request.putQueryParameter("CalledNumber", "1885xxxx");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(JSON.toJSONString(response.getData()));
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "762104B0-D58A-4D04-8486-7C5171D0D756",
  "Code": "OK",
  "CallId": "118720797955^10551795xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

5.5 取消点击呼叫

本章节为您介绍调用接口CancelCall取消点击呼叫接口ClickToDial发起的呼叫。

功能介绍

点击呼叫是指通过调用语音服务接口，通过语音服务分配的号码分别向主叫、被叫发起呼叫，双方接通后建立起正常通话，通话双方显示的号码均为语音服务平台号码。常用于企业办公电话等，例如钉钉办公电话。通过点击呼叫接口，可隐藏通话双方真实号码，同时平台可留存双方通话记录。

实现方法

调用该API实现取消点击呼叫功能之前，您需要成功调用ClickToDial实现点击呼叫，并获取CallId。

接口详情请参见[#unique_30](#)。

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class CancelCall {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        CommonRequest request = new CommonRequest();
        request.setMethod(MethodType.POST);
        request.setDomain("dyvmsapi.aliyuncs.com");
        request.setVersion("2017-05-25");
        request.setAction("CancelCall");
        request.putQueryParameter("CallId", "117059405036^10385912xx");
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

正常返回结果如下所示：

```
{
  "Status": "true",
  "Message": "OK",
  "RequestId": "D9CB3933-9FE3-4870-BA8E-2BEE91B69D23",
  "Code": "OK"
}
```

```
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

5.6 智能外呼

本章节为您介绍调用接口SmartCall发起智能语音交互通话的功能介绍和实现方法。

功能介绍

在智能语音交互通话中，支持使用控制台预设的机器人话术，或在每轮通话中通过回调接口返回业务方设置的回应方式。接口SmartCall需要和[#unique_32](#)联合使用，语音平台发起呼叫后，会把转换后的语音文本回传给业务方，业务方把下一步的执行动作返回给语音平台。

实现方法

调用API实现智能外呼功能之前，您需要完成以下操作：

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Java SDK](#)。
- 已在[服务开通页面](#)开通智能外呼ASR，并在[订阅回执消息页面](#)开启智能外呼ASR实时交互。详情请参见[回调接口说明](#)和[智能外呼回调HTTP接口](#)。

本章节调用接口详情，请参见[#unique_33](#)。

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class SmartCall {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);

        CommonRequest request = new CommonRequest();
        request.setMethod(MethodType.POST);
        request.setDomain("dyvmsapi.aliyuncs.com");
        request.setVersion("2017-05-25");
        request.setAction("SmartCall");
        request.putQueryParameter("CalledShowNumber", "0633676xx");
        request.putQueryParameter("CalledNumber", "10xx");
        request.putQueryParameter("VoiceCode", "3355eedd-3706-4f75-bff9-e645e88e1730.wav");
        request.putQueryParameter("PauseTime", "800");
        request.putQueryParameter("MuteTime", "10000");
        request.putQueryParameter("ActionCodeBreak", "false");
    }
}
```

```
try {
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "A90E4451-FED7-49D2-87C8-00700A8C4D0D",
  "Code": "OK",
  "CallId": "116012854210^10281427xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

5.7 查询通话记录详情

本章节为您介绍阿里云语音服务语音通知的功能介绍和实现方法，您可以通过调用QueryCallDetailByCallId查询通话记录详情。

功能介绍

当您调用接口成功实现语音通知、语音验证码、语音IVR、点击拨号、智能外呼功能，您可以调用QueryCallDetailByCallId根据接口返回的CallId查询某个时间点的通话记录。

实现方法

在调用API实现该功能前，您需要完成以下操作：

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Java SDK](#)。

本章节调用接口详情，请参见[#unique_35](#)。

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;

public class QueryCallDetailByCallId {
    public static void main(String[] args) {
```

```
DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou", "<accessKeyId>", "<accessSecret>");
IAcsClient client = new DefaultAcsClient(profile);

CommonRequest request = new CommonRequest();
request.setMethod(MethodType.POST);
request.setDomain("dyvmsapi.aliyuncs.com");
request.setVersion("2017-05-25");
request.setAction("QueryCallDetailByCallId");
request.putQueryParameter("CallId", "100625930001^10019107xx");
request.putQueryParameter("ProdId", "11000000300004");
request.putQueryParameter("QueryDate", "1577255564");
try {
    CommonResponse response = client.getCommonResponse(request);
    System.out.println(response.getData());
} catch (ServerException e) {
    e.printStackTrace();
} catch (ClientException e) {
    e.printStackTrace();
}
}
```

正常返回结果如下所示：

```
{
  "Data": {"caller": "1876713xx", "startDate": "", "stateDesc": "502", "duration": 0, "callerShowNumber": "0534475xx", "gmtCreate": "2019-12-24 20:09:06", "state": "502", "endDate": "", "calleeShowNumber": "0534475xx", "callee": "1373546xx"}
  "Message": "OK",
  "RequestId": "A5291B00-1036-4CA2-8BFD-1238B784C9BA",
  "Code": "OK"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6 Python示例

6.1 调用SDK

本章节为您介绍调用阿里云语音服务SDK的具体步骤，您可以参考本文档，调用其他API实现对应功能。

前提条件

在使用语音服务SDK之前，您需要：

- 开通语音服务SDK并购买语音号码，具体操作请参见[#unique_8](#)。
- 获取Accesskey信息，具体操作请参见[#unique_4](#)。
- 完成安装SDK，具体操作请参见[Python SDK](#)。



说明：

本章节以调用SingleCallByTts实现向指定号码发送语音验证码为您示例，API详情请参见[#unique_9](#)。

操作步骤

1. 创建client对象，并传入AccessKey信息。

```
client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
```



说明：

RegionId默认选择cn-hangzhou。

2. 创建request对象，并设置请求参数。

```
request = SingleCallByTtsRequest()
request.set_accept_format('json')
request.set_CalledShowNumber("1862xxxx")
request.set_CalledNumber("02131xxxx")
request.set_TtsCode("TTS_177553")
```

...

3. 创建response对象，并传入请求参数。

```
response = client.do_action_with_exception(request)
```

4. 打印输出信息。

```
print(str(response, encoding='utf-8'))
```

```
#!/usr/bin/env python
#coding=utf-8

from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkdysmsapi.request.v20170525.SingleCallByTtsRequest import
SingleCallByTtsRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = SingleCallByTtsRequest()
request.set_accept_format('json')
request.set_CalledShowNumber("1862xxxx")
request.set_CalledNumber("02131xxxx")
request.set_TtsCode("TTS_177553")

response = client.do_action_with_exception(request)
print(str(response, encoding='utf-8'))
```

返回示例如下所示。

- 正常返回示例。

```
{
  "Message": "OK",
  "RequestId": "2886A9EB-44B6-43C6-9E2F-031ECCCE4CEB",
  "Code": "OK",
  "CallId": "118697418557^1054..."
}
```

- 异常返回示例。

```
{
  "Message": "文本转语音模板不合法",
  "RequestId": "AE3D4BAE-F95C-4C0B-9C1E-DDF0F49C5645",
  "Code": "isv.TEMPLATE_ILLEGAL"
}
```

6.2 语音通知

本章节为您介绍阿里云语音服务语音通知的功能介绍和实现方法，您可以通过调用SingleCallByVoice实现发起语音文件类型的语音通知、调用SingleCallByTts实现发起文本转语音类型的语音通知。

功能介绍

语音通知可以很好的解决通知到达率延迟、关注度不高、阅读体验差等问题，并且您可以通过电话获得更全面的信息通知。您只需要提前设置通知模版内容，调用API就可以给用户推送语音通知，并可以设置播放音量、播放次数等。

实现方法

在实现调用API实现语音通知功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Python SDK](#)。

本章节调用接口参数详情，请参见[#unique_24](#)、[#unique_9](#)。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkdyvmsapi.request.v20170525.SingleCallByTtsRequest import SingleCallByTtsRequest
from aliyunsdkdyvmsapi.request.v20170525.SingleCallByVoiceRequest import SingleCallByVoiceRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')

request1 = SingleCallByVoiceRequest()
request1.set_accept_format('json')
request1.set_CalledShowNumber("18850505050")
request1.set_CalledNumber("15750505050")
request1.set_VoiceCode("e271f3f2-e155-4366-a9f4-0fe55765b3ec.wav")
response1 = client.do_action_with_exception(request1)
print('发起语音文件通知返回')
print(response1)

request2 = SingleCallByTtsRequest()
request2.set_accept_format('json')
request2.set_CalledShowNumber("18850505050")
request2.set_CalledNumber("15750505050")
request2.set_TtsCode("TTS_130555018")
request2.set_TtsParam("{\"name\": \"测试\", \"time\": \"2019年\"}")
response2 = client.do_action_with_exception(request2)
print('文本转语音类型返回')
print(response2)
```

正常返回结果如下所示：

```
发起语音文件通知返回
{
```

```
"Message": "OK",
"RequestId": "4ABF7FB6-449C-4624-8CC6-9AAE368453AA",
"Code": "OK",
"CallId": "118720962490^10551812xx"
}
文本转语音类型返回
{
"Message": "OK",
"RequestId": "762104B0-D58A-4D04-8486-7C5171D0D756",
"Code": "OK",
"CallId": "118720797955^10551795xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6.3 语音IVR

本章节为您介绍阿里云语音服务语音IVR的功能介绍和实现方法，您可以通过调用IvrCall实现交互式语音应答。

功能介绍

语音IVR是指通过调用语音呼叫的API，从运营商网络向指定号码发起一通呼叫，呼叫被接起后，播放一段指定音频，用户根据音频引导，通过手机按键信息返回意图，语音平台通过消息回执返回按键信息给企业业务系统。

实现方法

在实现调用API实现语音IVR功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Python SDK](#)。

本章节调用接口详情，请参见[#unique_27](#)。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkdysmsapi.request.v20170525.IvrCallRequest import IvrCallRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')

request = IvrCallRequest()
request.set_accept_format('json')
request.set_CalledShowNumber("05715621xx")
request.set_CalledNumber("1500000xx")
request.set_StartCode("TTS_713900000")
request.set_StartTtsParams("{\"product\": \"aliyun\", \"code\": \"123\"}")
request.set_MenuKeyMaps([
    {
        "Key": "1",
        "Code": "9a9d7222-670f-40b0-a3af.wav"
    }
],
```

```
{
  "Key": "2",
  "Code": "TTS_71390000",
  "TtsParams": "{\"product\": \"aliyun\", \"code\": \"123\"}"
}
])
request.set_PlayTimes(3)
request.set_Timeout(3000)
response = client.do_action_with_exception(request)
print(response)
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "4ABF7FB6-449C-4624-8CC6-9AAE368453AA",
  "Code": "OK",
  "CallId": "118720962490^10551812xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6.4 点击呼叫

本章节为您介绍如何调用ClickToDial实现阿里云语音通知点击呼叫功能。

功能介绍

点击呼叫是指通过调用语音服务接口，通过语音服务分配的号码分别向主叫、被叫发起呼叫，双方接通后建立起正常通话，通话双方显示的号码均为语音服务平台号码。常用于企业办公电话等，例如钉钉办公电话。通过点击呼叫接口，可隐藏通话双方真实号码，同时平台可留存双方通话记录。

实现方法

实现点击呼叫功能前，您需要完成以下操作。

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Python SDK](#)。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkdysmsapi.request.v20170525.ClickToDialRequest import ClickToDialRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = ClickToDialRequest()
request.set_accept_format('json')
request.set CallerShowNumber("123xxxx")
request.set CallerNumber("1575xxxx")
request.set CalledShowNumber("12xxxx")
request.set CalledNumber("1885xxxx")
response = client.do_action_with_exception(request)
```

```
print(response)
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "762104B0-D58A-4D04-8486-7C5171D0D756",
  "Code": "OK",
  "CallId": "118720797955^10551795xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6.5 取消点击呼叫

本章节为您介绍如何调用接口CancelCall取消点击呼叫接口ClickToDial发起的呼叫。

功能介绍

点击呼叫是指通过调用语音服务接口，通过语音服务分配的号码分别向主叫、被叫发起呼叫，双方接通后建立起正常通话，通话双方显示的号码均为语音服务平台号码。常用于企业办公电话等，例如钉钉办公电话。通过点击呼叫接口，可隐藏通话双方真实号码，同时平台可留存双方通话记录。

实现方法

调用该API实现取消点击呼叫功能之前，您需要成功调用ClickToDial实现点击呼叫，并获取CallId。

接口详情请参见[#unique_30](#)。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunsdkdysmsapi.request.v20170525.CancelCallRequest import CancelCallRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = CancelCallRequest()
request.set_accept_format('json')
request.set_CallId("117059405036^10385912xx")
response = client.do_action_with_exception(request)
print(response)
```

正常返回结果如下所示：

```
{
  "Status": "true",
  "Message": "OK",
  "RequestId": "D9CB3933-9FE3-4870-BA8E-2BEE91B69D23",
  "Code": "OK"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6.6 智能外呼

本章节为您介绍调用接口SmartCall发起智能语音交互通话的功能介绍和实现方法。

功能介绍

在智能语音交互通话中，支持使用控制台预设的机器人话术，或在每轮通话中通过回调接口返回业务方设置的回应方式。接口SmartCall需要和[#unique_32](#)联合使用，语音平台发起呼叫后，会把转换后的语音文本回传给业务方，业务方把下一步的执行动作返回给语音平台。

实现方法

调用API实现智能外呼功能之前，您需要完成以下操作：

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Python SDK](#)。
- 已在[服务开通页面](#)开通智能外呼ASR，并在[订阅回执消息页面](#)开启智能外呼ASR实时交互。详情请参见[回调接口说明](#)和[智能外呼回调HTTP接口](#)。

本章节调用接口详情，请参见[#unique_33](#)。

```
from aliynsdcore.client import AcsClient
from aliynsdcore.acs_exception.exceptions import ClientException
from aliynsdcore.acs_exception.exceptions import ServerException
from aliynsdkdvmsapi.request.v20170525.SmartCallRequest import SmartCallRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = SmartCallRequest()
request.set_accept_format('json')
request.set_CalledShowNumber("0633676xx")
request.set_CalledNumber("10xx")
request.set_VoiceCode("3355eedd-3706-4f75-bff9-e645e88e1730.wav")
request.set_PauseTime(800)
request.set_MuteTime(10000)
response = client.do_action_with_exception(request)
print(response)
```

正常返回结果如下所示：

```
{
  "Message": "OK",
  "RequestId": "A90E4451-FED7-49D2-87C8-00700A8C4D0D",
  "Code": "OK",
  "CallId": "116012854210^10281427xx"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

6.7 查询通话记录详情

本章节为您介绍阿里云语音服务语音通知的功能介绍和实现方法，您可以通过调用QueryCallDetailByCallId查询通话记录详情。

功能介绍

当您调用接口成功实现语音通知、语音验证码、语音IVR、点击拨号、智能外呼功能，您可以调用QueryCallDetailByCallId根据接口返回的CallId查询某个时间点的通话记录。

实现方法

在调用API实现该功能前，您需要完成以下操作：

- 购买语音号码并审核通过语音模板或语音文件，具体操作请参见[#unique_8](#)。
- 获取AccessKey信息，详情请参见[#unique_4](#)。
- 完成本地SDK环境搭建，详情请参见[Python SDK](#)。

本章节调用接口详情，请参见[#unique_35](#)。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException
from aliyunsdkcore.acs_exception.exceptions import ServerException
from aliyunskdyvmsapi.request.v20170525.QueryCallDetailByCallIdRequest import
QueryCallDetailByCallIdRequest

client = AcsClient('<accessKeyId>', '<accessSecret>', 'cn-hangzhou')
request = QueryCallDetailByCallIdRequest()
request.set_accept_format('json')
request.set_CallId("100625930001^10019107xx")
request.set_ProdId(11000000300004)
request.set_QueryDate(1577255564)
response = client.do_action_with_exception(request)
print(response)
```

正常返回结果如下所示：

```
{
  "Data":{"caller":"1876713xx","startDate":"","stateDesc":"502","duration":0,"callerShow
Number":"0534475xx","gmtCreate":"2019-12-24 20:09:06","state":"502","endDate":"","
calleeShowNumber":"0534475xx","callee":"1373546xx"}
  "Message": "OK",
  "RequestId": "A5291B00-1036-4CA2-8BFD-1238B784C9BA",
  "Code": "OK"
}
```

调用SDK返回错误码，详情请参见[#unique_25](#)。

7 消息回执

7.1 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。

7.2 .NET Demo

本章节为您介绍.NET语言获取回执消息示例方法。

下载SDK

获取回执消息依赖.NET语言的阿里云SDK核心库及dybaseapi，其中dybaseapi包用于拉取MNS消息。您需要下载阿里云以下SDK：

- [阿里云SDK核心库](#)
- [aliyun-net-sdk-dybaseapi](#)

示例方法

```
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.QueryTokenForMnsQueueResponse.QueryTokenForMnsQueue_MessageTokenDTO;

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

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

            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,
queueName, 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.GetAcResponse(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 Dictionary<string, QueryTokenForMnsQueue_MessageTokenDTO>();

```

```
readonly Dictionary<string, Queue> queueMap = new Dictionary<string, Queue>();

public QueryTokenForMnsQueue_MessageTokenDTO GetTokenByMessageType(
IAcsClient acsClient, String messageType)
{
    QueryTokenForMnsQueueRequest request = new QueryTokenForMnsQueueRequest
    {
        MessageType = messageType
    };
    QueryTokenForMnsQueueResponse queryTokenForMnsQueueResponse = acsClient
.GetAcResponse(request);
    QueryTokenForMnsQueue_MessageTokenDTO token = queryTokenForMnsQueu
eResponse.MessageTokenDTO;
    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,
mnsAccountEndpoint, 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.  
BatchReceiveMessage(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);  
}  
}  
}
```

7.3 PHP Demo

本章节为您介绍PHP语言获取回执消息示例方法。

下载SDK

获取回执消息依赖Client包和SDK包，您需要下载以下SDK：

- [openapi-sdk-php-client](#)
- [openapi-sdk-php](#)

示例方法

```
<?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);
```

7.4 Python Demo

本章节为您介绍Python语言获取回执消息示例方法。

下载SDK

获取回执消息依赖Python语言的阿里云SDK核心库及dybaseapi，其中dybaseapi包用于拉取MNS消息，您需要下载以下SDK：

- [阿里云SDK核心库](#)
- [aliyun-python-sdk-dybaseapi](#)

示例方法

```
#!/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
QueryTokenForMnsQueueRequest
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
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)
```

7.5 Go Demo

本章节为您介绍Go语言获取回执消息示例方法。

下载SDK

获取回执消息，您需要下载[alibaba-cloud-sdk-go](#)。

示例方法

```
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)
    }
}
}
```