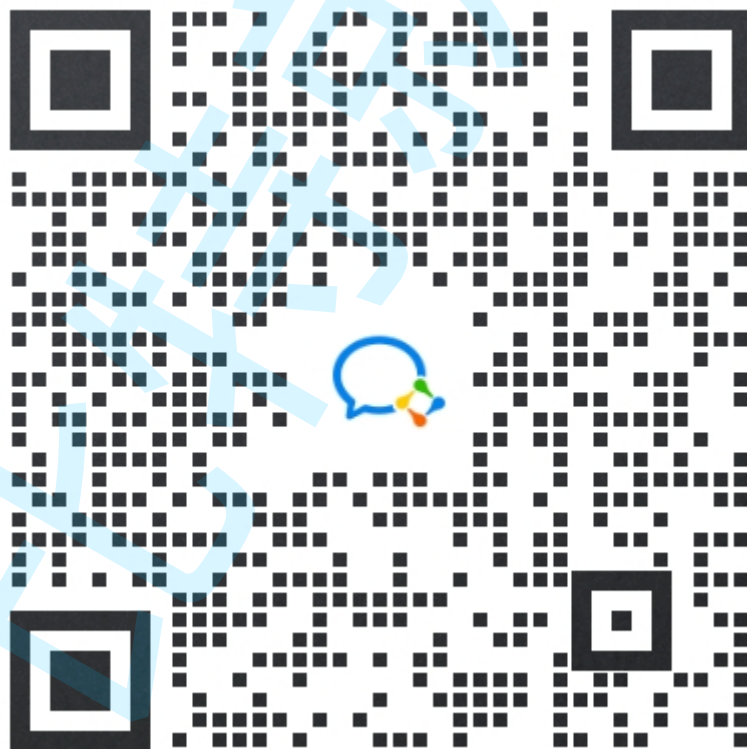


短信验证码 SDK

版权说明

本“比特就业课”项目（以下简称“本项目”）的所有内容，包括但不限于文字、图片、音频、视频、软件、程序、数据库、设计、布局、界面等，均由本项目的开发者或授权方拥有版权。我们鼓励个人学习者使用本项目进行学习和研究。在遵守相关法律法规的前提下，个人学习者可以下载、浏览、学习本项目的内容，并为了个人学习、研究或教学目的而使用其中的材料。但请注意，未经我们明确授权，个人学习者不得将本项目的内容用于任何商业目的，包括但不限于销售、转让、许可或以其他方式从中获利。此外，个人学习者也不得擅自修改、复制、传播、展示、表演或制作本项目内容的衍生作品。任何未经授权的使用均属侵权行为，我们将依法追究法律责任。如果您希望以其他方式使用本项目的内容，包括但不限于引用、转载、摘录、改编等，请事先与我们联系，获取书面授权。感谢您对“比特就业课”项目的关注与支持，我们将持续努力，为您提供更好的学习体验。特此说明。比特就业课版权所有方

对比特项目感兴趣，可以联系这个微信。



代码 & 板书链接

<https://gitee.com/bitedu-tech/cpp-chatsystem>

环境搭建

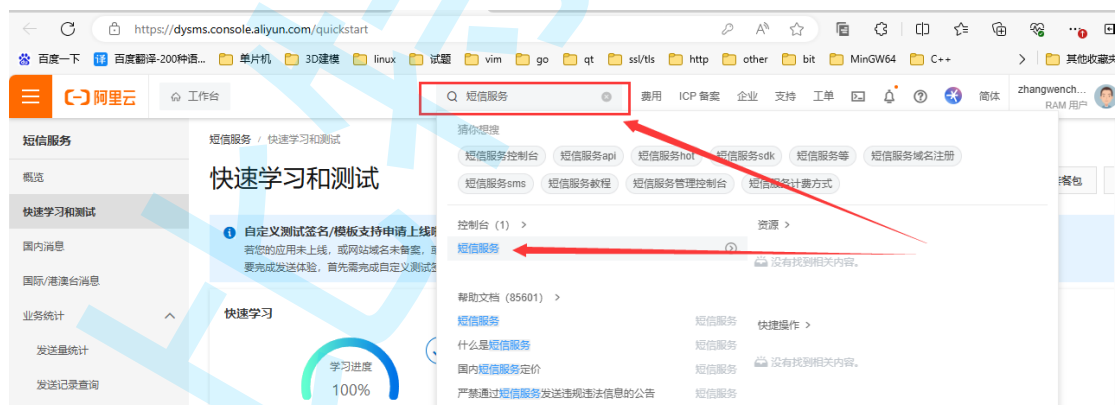
1. 注册/登录阿里云网站

网址：

https://account.aliyun.com/login/login.htm?oauth_callback=https%3A%2F%2Fdysms.console.aliyun.com%2F&lang=zh



2. 开通并购买短信服务



阿里云官网中搜索短信服务进入下面界面。



购买短信服务



进入购买页面



3. 开始学习和测试



按照步骤逐步进行即可：

1. 新增资质

← 新增资质

① 审核时长一般2小时内完成，涉及政企资质一般2个工作日内完成，近期平均完成审核时长约1小时，如遇升级核验、审核任务较多、非工作时间，审核时间可能会延长，请耐心等待。审核时间：周日9:00-21:00（法定节假日顺延）

基础信息

* 申请用途

☒ 自用
资质中企业或个人名称为本账号实名认证的名称
自用资质申请所需材料清单

☐ 他用
资质中企业或个人名称为非本账号实名认证的名称
他用资质申请所需材料清单

* 资质命名

为便于签名申请时搜索并关联正确的资质，请为资质命名（不要重复）
请输入中文、英文或数字，请不要重复。请不要输入纯数字名称

自己命名即可
主要是为了有多个资质的时候便于区分

备注

请输入内容

证件信息

身份证

* 上传

+

身份证人像面照片

+

身份证国徽面照片

+

手持身份证照片
请展示出身份证人像面

文件格式：支持jpg、png、gif、jpeg格式的图片，每张图片不大于5MB

更多资料

更多资料

更多资料

提交申请

取消

审核时长一般 2 小时内完成，涉及政企资质一般 2 个工作日内完成，近期平均完成审核时长约 1 小时。

2. 申请签名

← 添加签名

ⓘ 若个人客户，未上线应用或未备案域名建议使用[快速学习-发送测试](#)或[升级企业账号](#)。为了进一步管控违法违规短信的发送，已参考运营商实名发送短信要求，在中心增加实名认证环节。

* 签名

请填写真实应用名/网站名/公司名，长度限制在2-12个字符

0/12

签名不区分大小写，请勿填写“测试”字符

* 适用场景

☒ 通用 ⓘ

* 签名用途

☒ 自用（签名为本账号实名认证的网站、App等）

☐ 他用（签名为非本账号实名认证的企业、网站、产品名等）

* 资质信息

zsc / 审核通过

没有需要的资质，点击[添加资质](#)，点击查看[资质列表](#)。

* 签名来源

测试或学习

当前只能选择这个，其他的需要更进一步的资质

适用于应用开发测试阶段，但仅支持对绑定测试手机号码范围发送短信

场景说明

请填写场景说明，可提升通过率

0/500

更多资料

上传资料

可上传带有链接的登录页面、后台页面截图、软著、协议补充等资料
若多个资料可拼接成一个文件，支持png、jpg、jpeg、doc、docx、pdf格式。

ⓘ 审核时长：一般2小时内审核完成，且平均完成审核时效约1小时。如遇升级核验或审核任务较多时，审核时间可能会延长，请您耐心等待。
审核工作时间：周一至周日9:00-21:00（法定节日顺延）

提交 取消

3. 申请模板：

快速学习

学习进度

100%

新增资质

资质是申请签名的实名化信息，可复用

[前往查看](#)

申请签名

签名是短信中能代表发送者属性的字段。

[前往查看](#)

申请模板

模板即具体发送的短信内容。

[前往查看](#)

系统设置

发送短信前，请提前完成系统设置。

[前往查看](#)

发送短信

可通过控制台群发助手或API调用进行短信发送

[前往查看](#)

发送测试

API发送测试

控制台发送测试

1. 建议测试前确保套餐余额或账户资金充足，以保证发送测试成功。[去充值](#)

2. 本功能仅适用于API/控制台发送测试阶段，建议您尽早申请符合业务真实场景的签名、模板。

3. 需要添加测试手机号码，才能发送测试。您可以前往[业务统计](#)查看发送结果及明细。

1. 绑定测试手机号码

每个账号限制授权5个测试手机号码。

绑定测试手机号码

测试或学习只能发送短信给绑定的手机号
需要先绑定测试手机

选择测试方式

序号	手机号码	添加时间	操作
1		2024-06-05 14:51:07	删除
2		2024-05-29 18:07:09	删除
3		2024-05-06 16:37:22	删除
4		2024-04-01 11:32:46	删除

2. 发送测试

1

若您需要自定义内容的测试签名模板，先进行申请，审核通过后可适用。[申请自定义测试签名](#) [申请自定义测试模板](#)

产品类型

国内短信

签名

请选择签名

选择之前申请的签名

没有需要的签名，马上[添加签名](#)。点此[查看签名列表](#)

模板

全部

请选择模板

选择之前创建的模板

没有需要的模板，马上[添加模板](#)。点此[查看模板列表](#)

接收号码

请输入正确的手机号

输入绑定的手机号

如需号码不受发送频率限制，可[添加号码白名单](#)

发送短信

查看API Demo

计委说明：当前0字，预计发送条数约为0条（实际发送时，签名、变量长度会影响总字数）。[查看短信计算规则](#)

API 使用测试

- 创建 ACESSKey



AccessKey ID 和 AccessKey Secret 是我们访问阿里云 API 的密钥，具有该账户完全的权限，需要我们妥善保管。



点击创建并保存创建好的 AccessKey ID 和 AccessKey Secret，后边写代码的时候需要使用。

- 回到短信服务首页，调用 API 发送短信

快速学习

学习进度
100%

新增资质

资质是申请签名的实名化信息，可复用

前往查看

申请签名

签名是短信中能代表发送者属性的字段。

前往查看

申请模板

模板即具体发送的短信内容。

前往查看

系统设置

发送短信前，请提前完成系统设置。

前往查看

发送短信

可通过控制台群发助手或API调用进行短信发送

前往查看

发送测试

API发送测试

控制台发送测试

1. 建议测试前确保套餐包余量或账户资金充足，以保证短信测试成功。去充值

2. 本功能仅适用于API/控制台发送测试阶段，建议您尽早申请符合您业务真实场景的签名、模板。

3. 需要添加测试手机号码，才能发送测试。您可以前往 业务统计 查看发送结果及明细。

1. 绑定测试手机号码

每个账号限制授权5个测试手机号码。

绑定测试手机号码

序号	手机号码	添加时间	操作
1	15000000000	2024-06-05 14:51:07	删除
2	15000000000	2024-05-29 18:07:09	删除
3	15000000000	2024-05-06 16:37:22	删除
4	15000000000	2024-04-01 11:32:46	删除

2. 调用测试签名模板API

* 测试签名/模板类型

☒ 【自定义】测试签名/模板

可自定义内容的测试签名模板，需经审核但要求低些，仅支持验证码和通知短信申请

☐ 【专用】测试签名/模板

指定内容的测试签名模板，无需审核直接使用

* 签名

bitejiuyeye

没有自定义测试签名？马上申请自定义测试签名

* 模板

全部

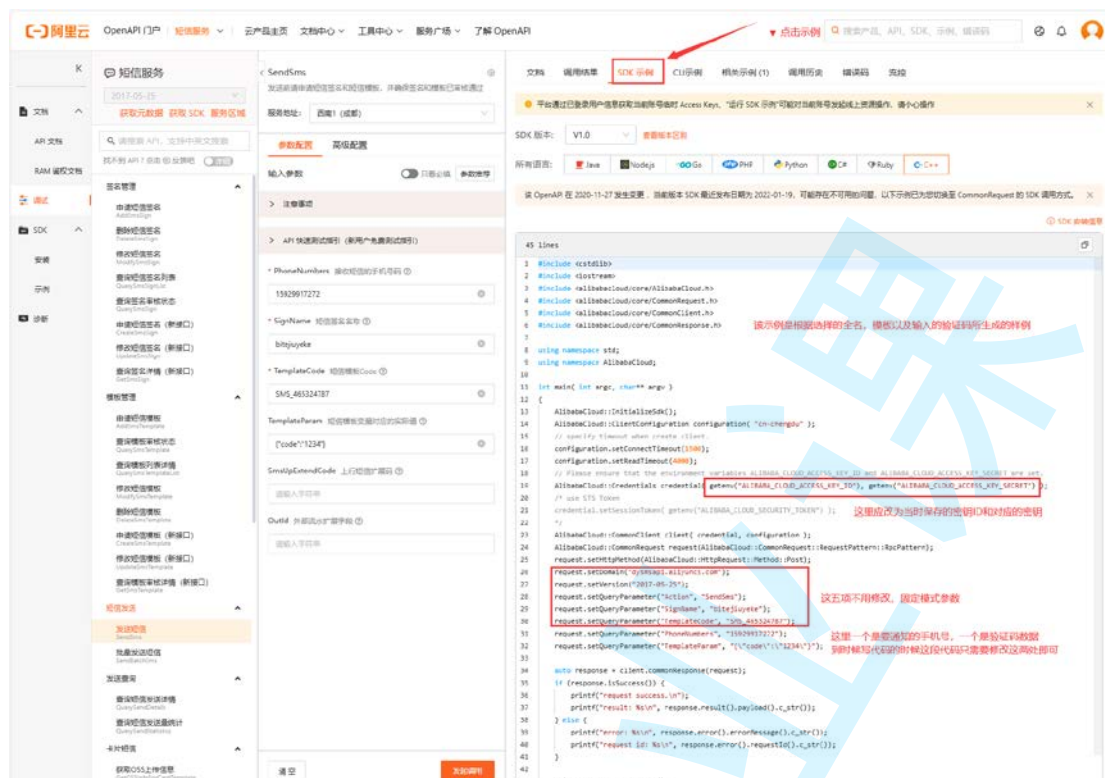
验证码

没有自定义测试模板？马上申请自定义测试模板

调用API发送短信

会自动跳到下面的界面，我们可以点击发起调用进行测试。

我们可以点击 SDK 示例代码，会有各种语言调研阿里云短信 API 的代码，我们直接复制即可，但是需要安装好对应的 SDK 组件。



集成短信 sdk 到项目

<https://help.aliyun.com/zh/sms/developer-reference/sdk-product-overview/?spm=a2c4g.11186623.0.0.6b6f12f3dvtugg>

SDK版本说明

阿里云短信服务OpenAPI门户为用户提供V1.0 SDK和V2.0 SDK。V2.0 SDK不涉及接口功能变更，主要是为了提升SDK的健壮性及使用体验。短信服务V1.0 SDK已不再维护，我们强烈推荐您使用V2.0 SDK。

- 说明
 - V1.0 SDK即原版SDK，V2.0 SDK即升级版SDK，版本对比详情请参见V1.0 SDK和V2.0 SDK。
 - 2019年01月22日，短信服务发布了V2.0 SDK。此日期后接入SDK的用户所使用的均为V2.0 SDK。

各语言SDK下载

短信服务V2.0 SDK源码已托管至开源平台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。您也可使用CommonRequest方式进行调用，更多关于CommonRequest调用方式请参见泛化调用与特化调用。更多示例代码可通过OpenAPI 开发者门户获取。

主流语言	Github地址	最新版SDK下载地址
Java	java-dysmsapi-20170525	下载Java SDK
Python	python-dysmsapi-20170525	下载Python SDK
PHP	php-dysmsapi-20170525	下载PHP SDK
Go	go-dysmsapi-20170525	下载Go SDK
Node.js	nodejs-dysmsapi-20170525	下载Node.js SDK
C++	cpp-dysmsapi-20170525	下载C++ SDK
C#	csharp-dysmsapi-20170525	下载C# SDK
Swift	swift-dysmsapi-20170525	下载Swift SDK

这里有各个版本、语言的 SDK，我们按照需要下载即可。下面以 C++ SDK 为例，这里我们使用 1.0 版本的 SDK，因为 2.0 版本的 SDK 暂时不支持 C++。

短信服务

当前C++只有1.0版本支持

API 版本: 2017-05-25

SDK 版本: V1.0

[查看版本区别](#)

所有语言:

Java

Go

PHP

Python

C++

选择C++SDK

[快速入门](#)[进阶文档](#)[SDK 示例](#)[历史版本](#)

概述

环境要求

发布地址

源码仓库地址

安装方式

示例背景

完整代码示例

步骤介绍

概述

文档中 SDK 关于 API 的示例代码仅供参考, 各 API 的完整使用步骤与说明请参见[SDK 示例](#)和[OpenAPI 文档](#)。build passing  93%

环境要求

- 安装支持 C++ 11 或更高版本的编译器:
 - Windows: Visual Studio 2015 或以上版本
 - Linux: GCC 4.9 或以上版本
- 安装 CMake 3.0 或以上版本
- 建议 4G 或以上内存
- 安装 SDK 核心库 Core, Linux 下, 安装命令如下:

```
git clone https://github.com/aliyun/aliyun-openapi-cpp-sdk.git
cd aliyun-openapi-cpp-sdk
sudo sh easyinstall.sh core
```

- 安装阿里云 C++ SDK

C++

安装依赖

```
sudo apt-get install libcurl4-openssl-dev libssl-dev uuid-dev
libjsoncpp-dev
```

克隆代码

```
git clone https://github.com/aliyun/aliyun-openapi-cpp-sdk.git
```

一键编译安装

```
sudo sh easyinstall.sh core
```

Install the project...

```
- Install configuration: "Release"
- Installing: /usr/include/alibabacloud/core/AlibabaCloud.h
- Installing: /usr/include/alibabacloud/core/AsyncCallerContext.h
- Installing: /usr/include/alibabacloud/core/ClientConfiguration.h
- Installing: /usr/include/alibabacloud/core/CommonClient.h
- Installing: /usr/include/alibabacloud/core/CommonRequest.h
- Installing: /usr/include/alibabacloud/core/CommonResponse.h
- Installing: /usr/include/alibabacloud/core/Config.h
- Installing: /usr/include/alibabacloud/core/CoreClient.h
- Installing: /usr/include/alibabacloud/core/CoreExport.h
- Installing: /usr/include/alibabacloud/core/Credentials.h
- Installing: /usr/include/alibabacloud/core/CredentialsProvider.h
- Installing: /usr/include/alibabacloud/core/EcsMetadataFetcher.h
- Installing: /usr/include/alibabacloud/core/EndpointProvider.h
- Installing: /usr/include/alibabacloud/core/Error.h
- Installing: /usr/include/alibabacloud/core/Global.h
- Installing: /usr/include/alibabacloud/core/HmacSha1Signer.h
- Installing: /usr/include/alibabacloud/core/HttpClient.h
```

- 拷贝 SDK 实例代码到 C++源文件中

```
C++
#include <cstdlib>
#include <iostream>
#include <alibabacloud/core/AlibabaCloud.h>
#include <alibabacloud/core/CommonRequest.h>
#include <alibabacloud/core/CommonClient.h>
#include <alibabacloud/core/CommonResponse.h>

using namespace std;
using namespace AlibabaCloud;

int main( int argc, char** argv )
{
    AlibabaCloud::InitializeSdk();
    AlibabaCloud::ClientConfiguration configuration( "cn-
hangzhou" );
    // specify timeout when create client.
    configuration.setConnectTimeout(1500);
    configuration.setReadTimeout(4000);
    // 别忘了这里修改为你自己的密钥 ID 和密钥
    AlibabaCloud::Credentials
    credential( getenv("ALIBABA_CLOUD_ACCESS_KEY_ID"),
    getenv("ALIBABA_CLOUD_ACCESS_KEY_SECRET" ) );
    /* use STS Token

    credential.setSessionToken( getenv("ALIBABA_CLOUD_SECURITY_TOKEN")
    );
    */
```

```

        AlibabaCloud::CommonClient client( credential,
configuration );
        AlibabaCloud::CommonRequest
request(AlibabaCloud::CommonRequest::RequestPattern::RpcPattern);

request.setHttpMethod(AlibabaCloud::HttpRequest::Method::Post);
request.setDomain("dysmsapi.aliyuncs.com");
request.setVersion("2017-05-25");
request.setQueryParameter("Action", "SendSms");
request.setQueryParameter("SignName", "bitejiuyeye");
request.setQueryParameter("TemplateCode", "SMS_465324787");
request.setQueryParameter("PhoneNumbers", "18392852312");
request.setQueryParameter("TemplateParam",
"{\"code\": \"1234\"}");

        auto response = client.commonResponse(request);
        if (response.isSuccess()) {
            printf("request success.\n");
            printf("result: %s\n",
response.result().payload().c_str());
        } else {
            printf("error: %s\n",
response.error().errorMessage().c_str());
            printf("request id: %s\n",
response.error().requestId().c_str());
        }

        AlibabaCloud::ShutdownSdk();
        return 0;
}

```

编译代码， 链接阿里云 SDK 库文件

```

Shell
g++ -o main main.cc -std=c++11 -lalibabacloud-sdk-core

```

运行代码， 可以看到已经发送短信成功了。

```

root@hcss-ecs-2618:/home/zsc/third/aliyun_test# ./main
request success.
result: {"Message": "OK", "RequestId": "4ADD68DA-F16E-557C-BD76-F15DC57C4137", "Code": "OK", "BizId": "301121612642791620^0"}

```



阿里云 SDK 仓库: <https://github.com/alibaba/openapi-cpp-sdk/blob/master/README-CN.md>

封装思想:

在项目中, 我们实际上并不关心如何调用阿里云的 SDK, 我只是期望有一个接口能够设定手机号和验证码之后, 调用就进行发送, 因此我们可以将 SDK 中使用到的接口和对象进行二次封装, 简化项目中的使用:

- 将 SDK 中的 client 对象管理起来
- 向外提供一个向指定手机号发送指定短信的接口