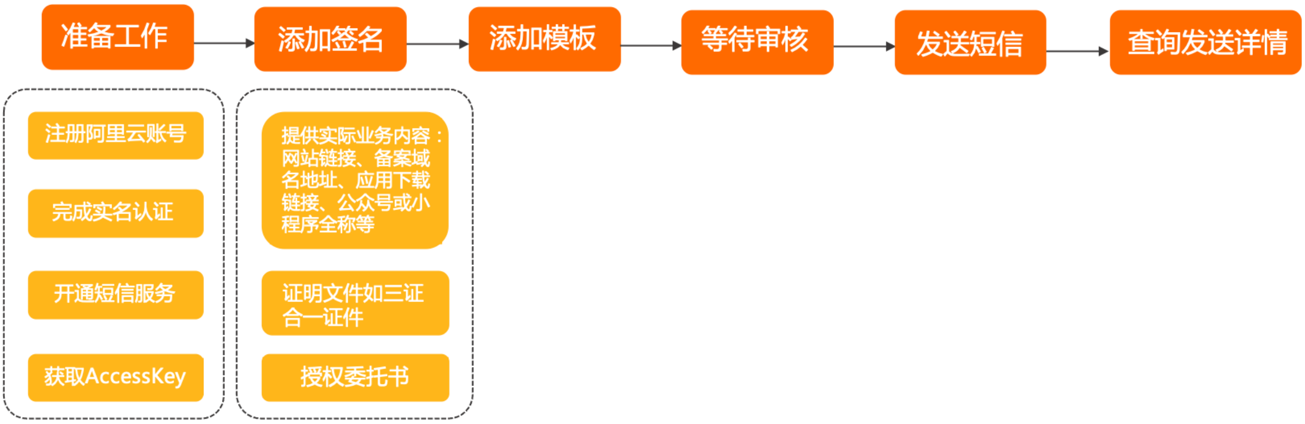
阿里云SMS

使用参考：https://help.aliyun.com/product/44282.html?spm=a2c4g.44282.0.0.6e7a6d7e25PM4X

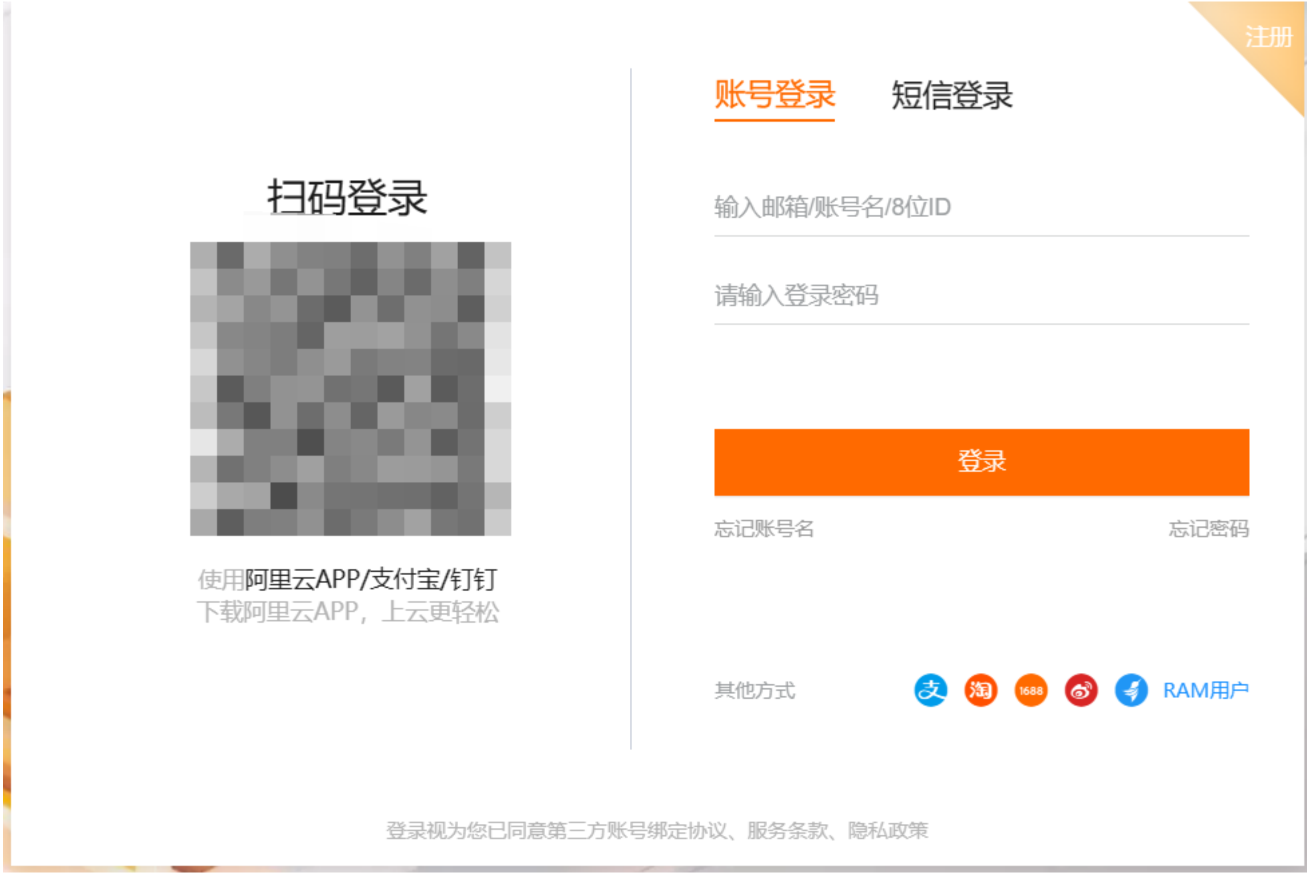
整体使用流程



一、账号相关

1 注册阿里云

进入控制台首页：https://home.console.aliyun.com/

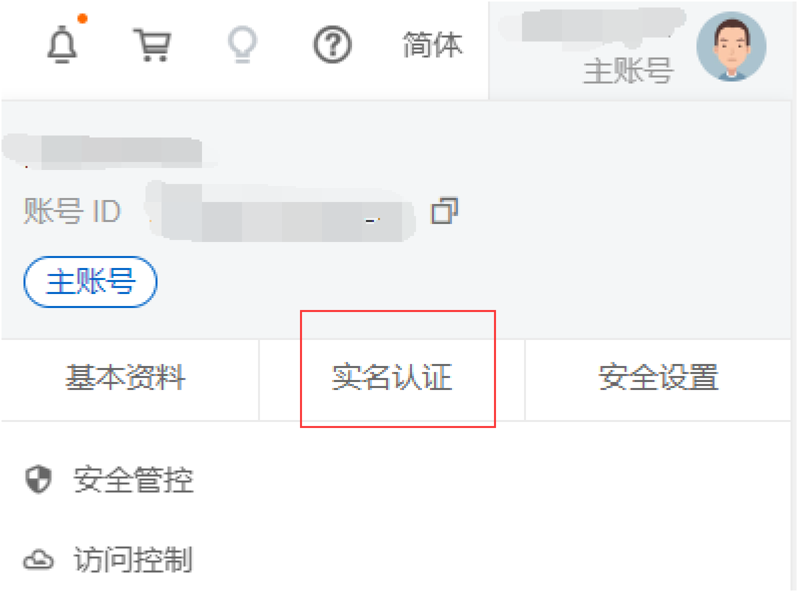


登录视为您已同意第三方账号绑定协议、服务条款、隐私政策

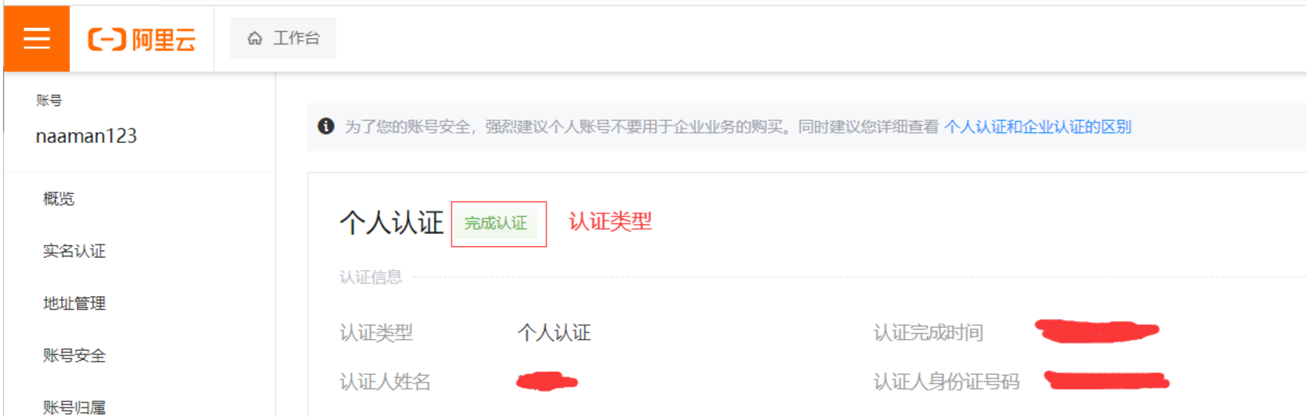
直接使用支付宝或钉钉扫描登录即可注册和完成实名认证进入你的控制台。

2 完成实名认证

将鼠标移动到控制台右上角账号图标上面，在弹出框找到实名认证入口，示意图如下图所示



填写对应资料完成实名认证，认证成功的账号可以在实名认证界面看到类似下面的效果



为了您的账号安全，强烈建议个人账号不要用于企业业务的购买。同时建议您详细查看 个人认证和企业认证的区别

个人认证 完成认证 认证类型

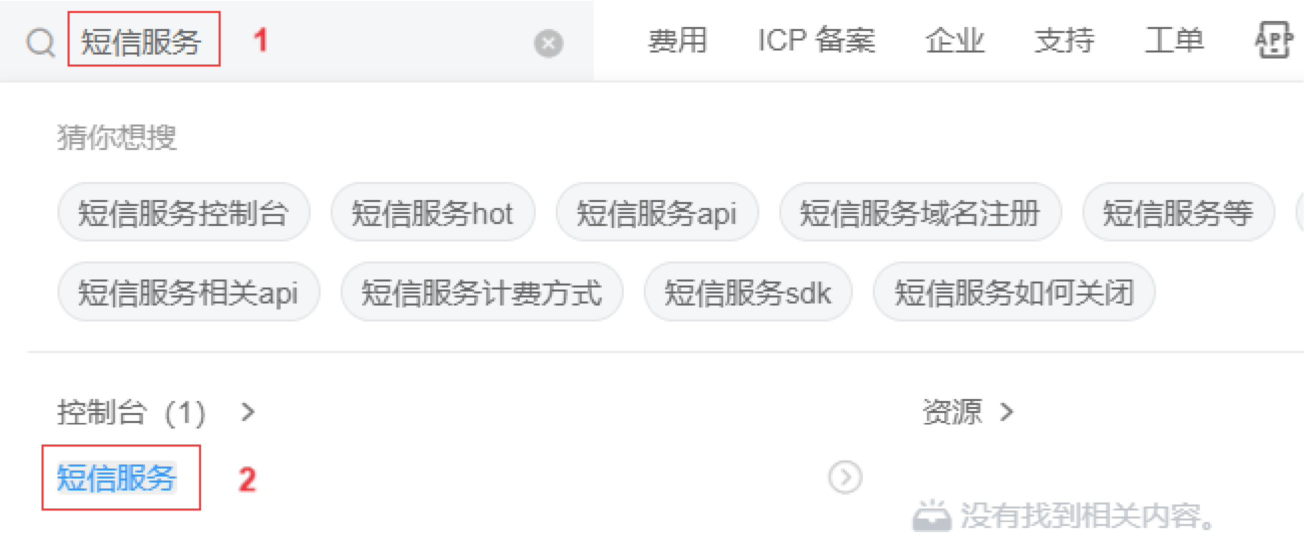
认证信息

认证类型 个人认证 认证完成时间 [redacted]

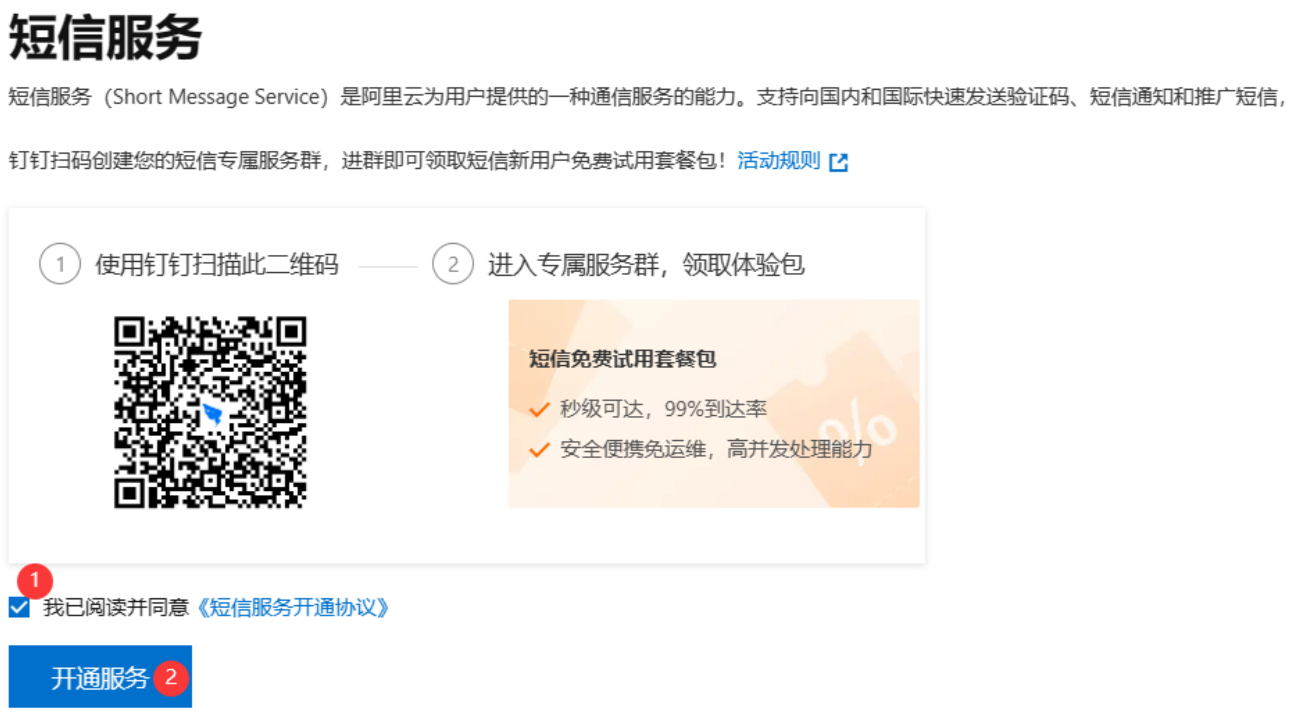
认证人姓名 [redacted] 认证人身份证号码 [redacted]

3 开通短信服务

在控制台中搜索短信服务，搜索示意效果如下图所示。



点击短信服务，进入开通确认界面



短信服务

短信服务（Short Message Service）是阿里云为用户提供的一种通信服务的能力。支持向国内和国际快速发送验证码、短信通知和推广短信，

钉钉扫码创建您的短信专属服务群，进群即可领取短信新用户免费试用套餐包！活动规则

1 使用钉钉扫描此二维码 —— 2 进入专属服务群，领取体验包

短信免费试用套餐包

✓ 秒级可达，99%到达率

✓ 安全便携免运维，高并发处理能力

1 ☑ 我已阅读并同意《短信服务开通协议》

按照上面的步骤操作即可

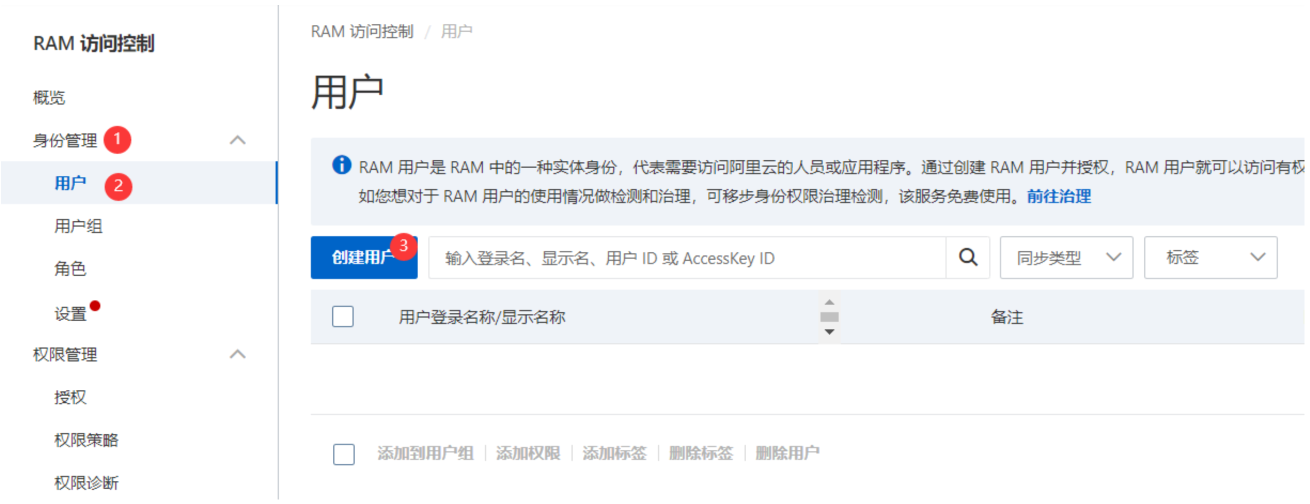
4 获取AccessKey

操作步骤参考：https://help.aliyun.com/zh/ram/user-guide/create-an-accesskey-pair

由于主账号的AccessKey功能将要下线，所以要使用子账号方式创建key，首先登录RAM控制台

找到用户列表，如果用户列表为空，那么需要创建用户，创建步骤如下：

打开用户创建页面



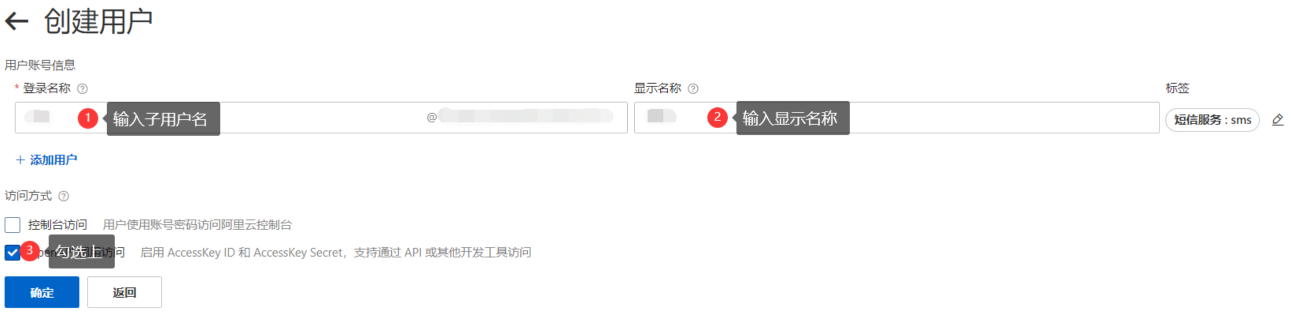
RAM 用户是 RAM 中的一种实体身份，代表需要访问阿里云的人员或应用程序。通过创建 RAM 用户并授权，RAM 用户就可以访问有权

如您想对于 RAM 用户的使用情况做检测和治理，可移步身份权限治理检测，该服务免费使用。前往治理

☐ 用户登录名称/显示名称	备注

☐ 添加到用户组 | 添加权限 | 添加标签 | 删除标签 | 删除用户

输入用户信息



← 创建用户

用户账号信息

* 登录名称 [redacted] @[redacted]

显示名称 [redacted]

标签 短信服务 : sms

＋ 添加用户

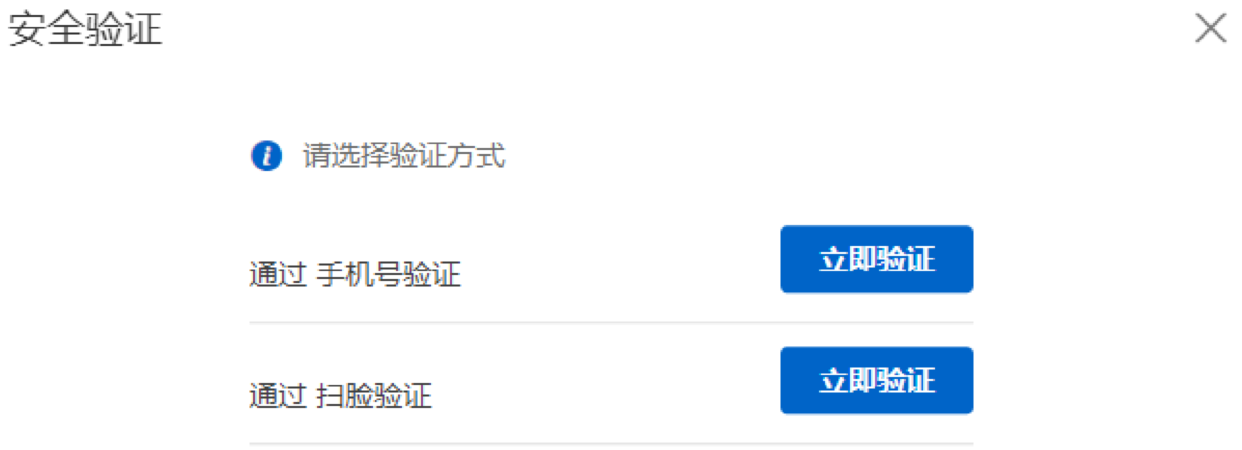
访问方式

□ 控制台访问 用户使用账号密码访问阿里云控制台

☑ OpenAPI 调用访问 启用 AccessKey ID 和 AccessKey Secret，支持通过 API 或其他开发工具访问

确定 返回

点击确认后，需要验证



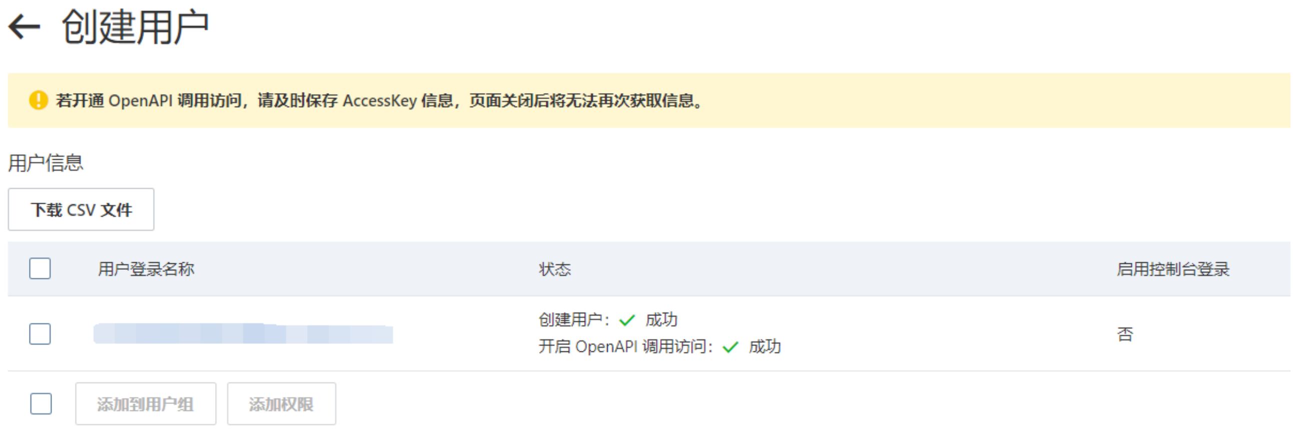
安全验证

请选择验证方式

通过 手机号验证 立即验证

通过 扫脸验证 立即验证

自己选择一种验证方式即可，验证通过后可以看到你的用户列表



← 创建用户

若开通 OpenAPI 调用访问，请及时保存 AccessKey 信息，页面关闭后将无法再次获取信息。

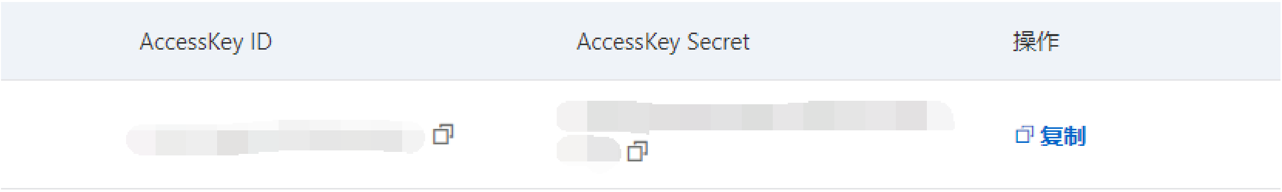
用户信息

下载 CSV 文件

□	用户登录名称	状态	启用控制台登录
□	[redacted]	创建用户: ✓ 成功 开启 OpenAPI 调用访问: ✓ 成功	否

□ 添加到用户组 添加权限

在用户的最后列的位置已经有对应的key了，如下图所示

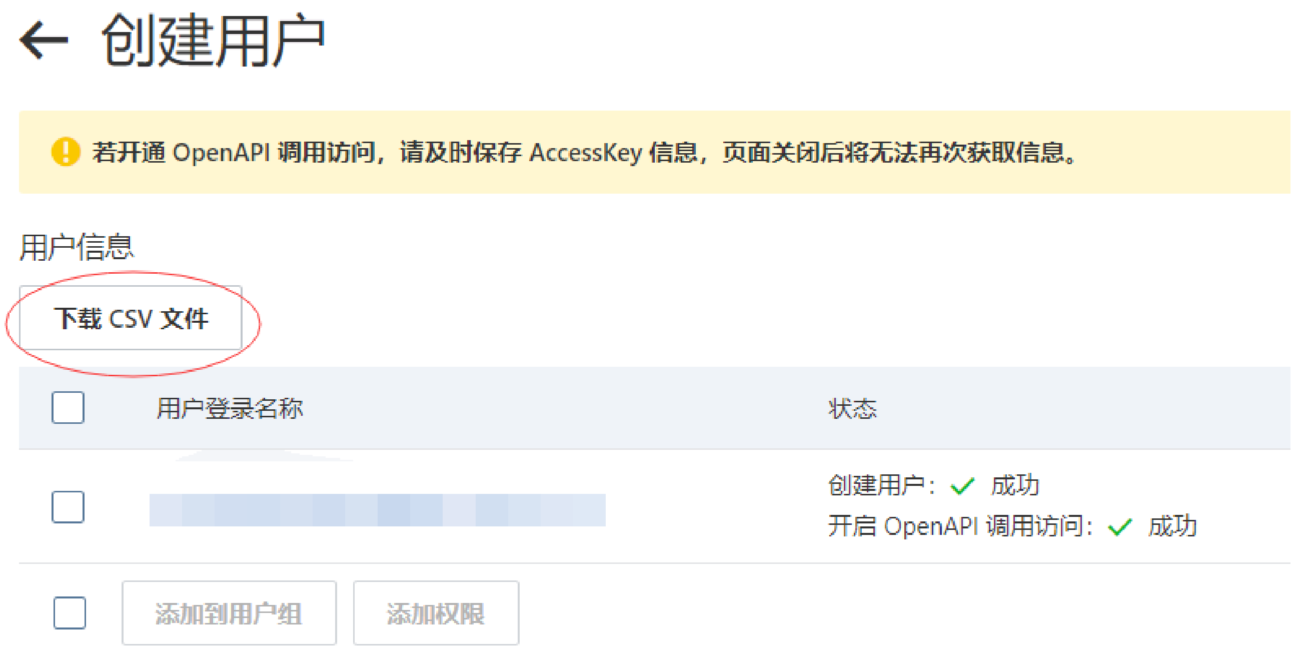


AccessKey ID	AccessKey Secret	操作
[redacted]	[redacted]	复制

注意：

- RAM用户的`AccessKey Secret`只在创建时显示，不支持查看，请妥善保管。
- 如果`AccessKey`泄露或丢失，请删除并创建新的`AccessKey`。每个RAM用户最多允许创建2个`AccessKey`。

你可以通过下载csv文件保存key到本地方便后续使用



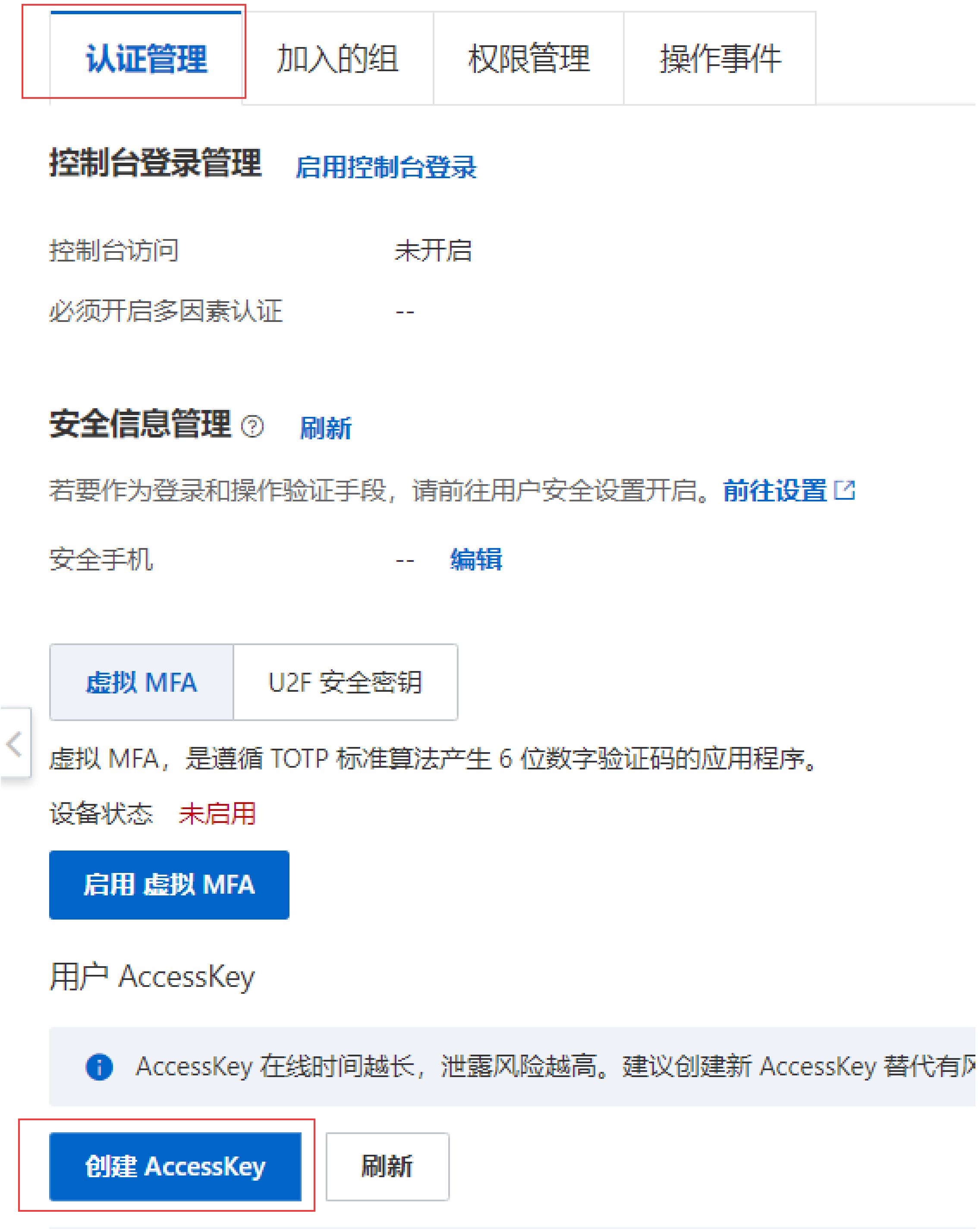
← 创建用户

若开通 OpenAPI 调用访问，请及时保存 AccessKey 信息，页面关闭后将无法再次获取信息。

用户信息

□	用户登录名称	状态
□	[redacted]	创建用户：✓ 成功 开启 OpenAPI 调用访问：✓ 成功

如果你要为已有用户添加key，直接点击用户列表中对应用户，然后在找到创建`AccessKey`按钮即可创建，如下图所示



认证管理 | 加入的组 | 权限管理 | 操作事件

控制台登录管理 启用控制台登录

控制台访问 未开启

必须开启多因素认证 --

安全信息管理 ⓘ 刷新

若要作为登录和操作验证手段，请前往用户安全设置开启。前往设置

安全手机 -- 编辑

虚拟 MFA | U2F 安全密钥

虚拟 MFA，是遵循 TOTP 标准算法产生 6 位数字验证码的应用程序。

设备状态 未启用

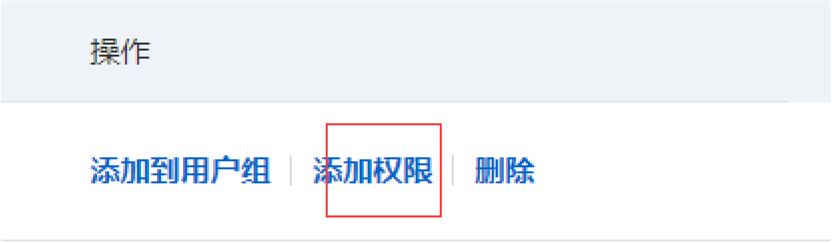
启用 虚拟 MFA

用户 AccessKey

AccessKey 在线时间越长，泄露风险越高。建议创建新 AccessKey 替代有

创建 AccessKey 刷新

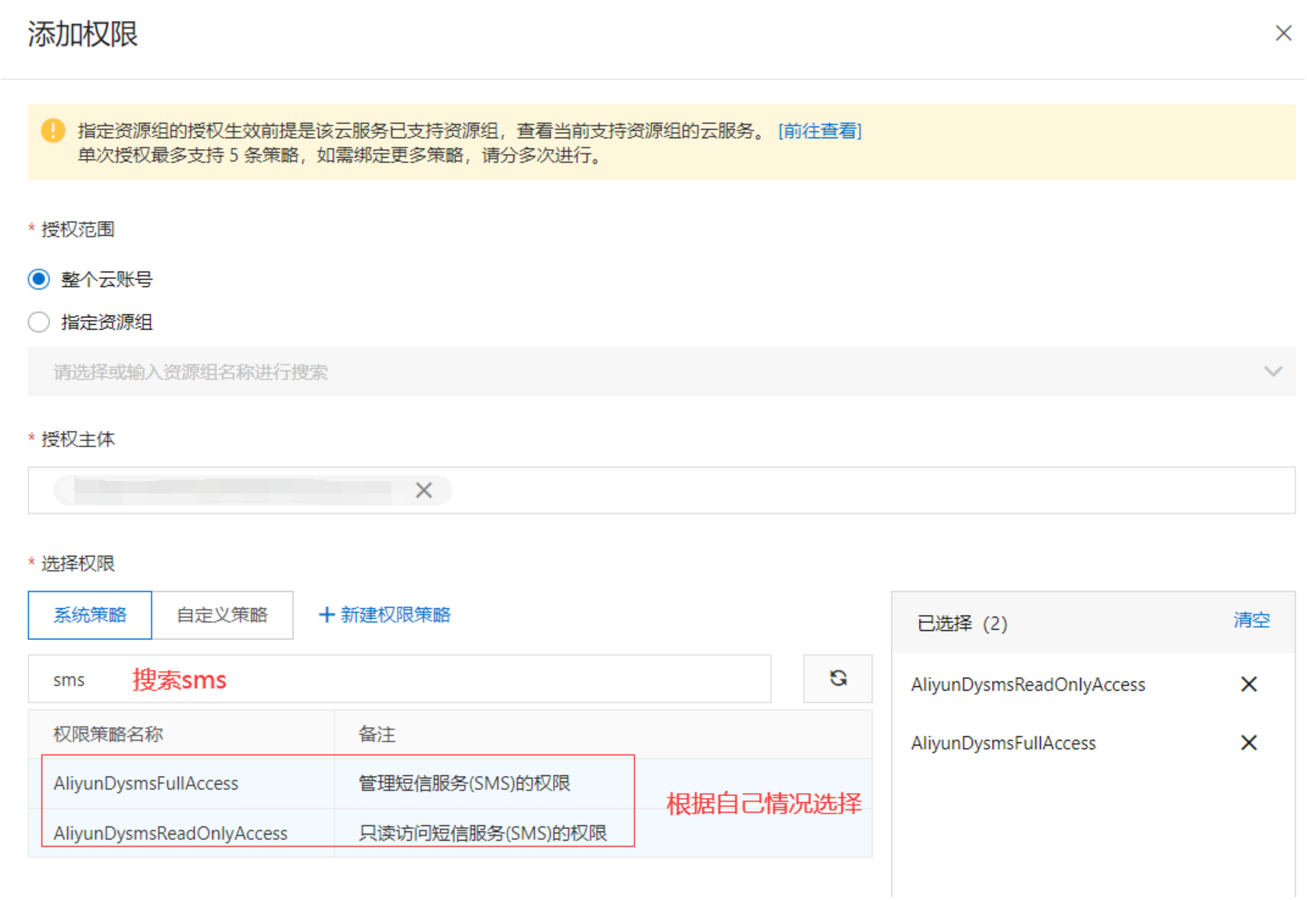
添加短信服务权限，在对应用户列表后面，找到操作菜单中的添加权限，如下图所示



操作

添加到用户组 | 添加权限 | 删除

添加短信服务权限



添加权限

ⓘ 指定资源组的授权生效前提是该云服务已支持资源组，查看当前支持资源组的云服务。[前往查看]
单次授权最多支持 5 条策略，如需绑定更多策略，请分多次进行。

* 授权范围

● 整个云账号

○ 指定资源组

请选择或输入资源组名称进行搜索

* 授权主体

[redacted] ×

* 选择权限

sms 搜索sms

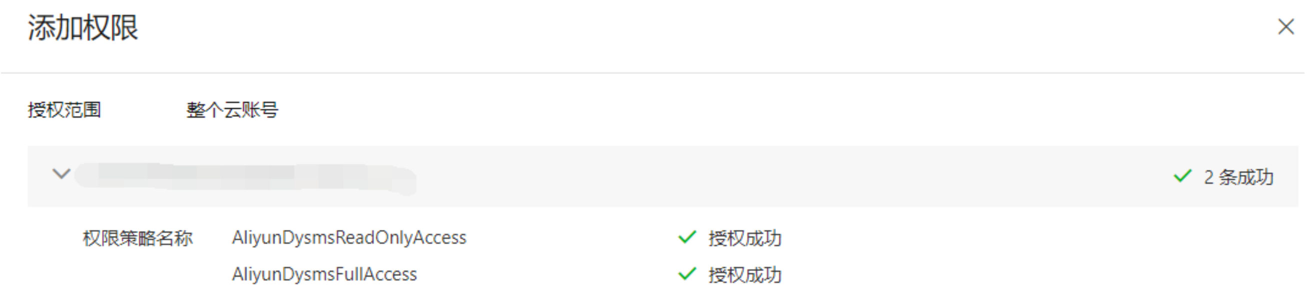
权限策略名称	备注
AliyunDysmsFullAccess	管理短信服务(SMS)的权限
AliyunDysmsReadOnlyAccess	只读访问短信服务(SMS)的权限

根据自己情况选择

已选择（2） 清空

- AliyunDysmsReadOnlyAccess ×
- AliyunDysmsFullAccess ×

点击确认即可，授权成功效果如下图



添加权限

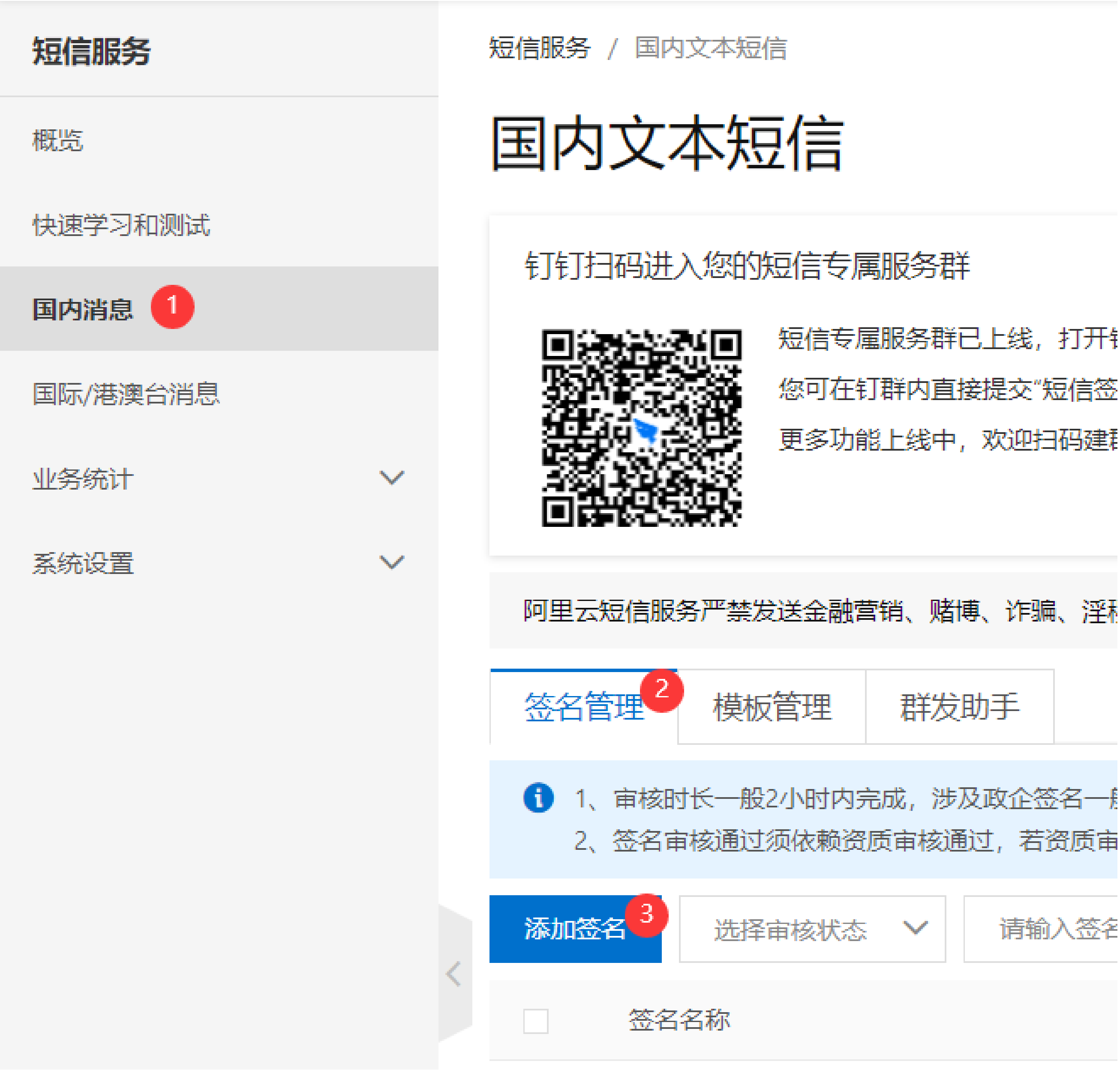
授权范围 整个云账号

[redacted] ✓ 2 条成功

权限策略名称		
	AliyunDysmsReadOnlyAccess	✓ 授权成功
	AliyunDysmsFullAccess	✓ 授权成功

5 添加短信签名

在短信服务控制台中找到签名管理，如下图所示



钉钉扫码进入您的短信专属服务群

短信专属服务群已上线，打开钅

您可在钉群内直接提交“短信签

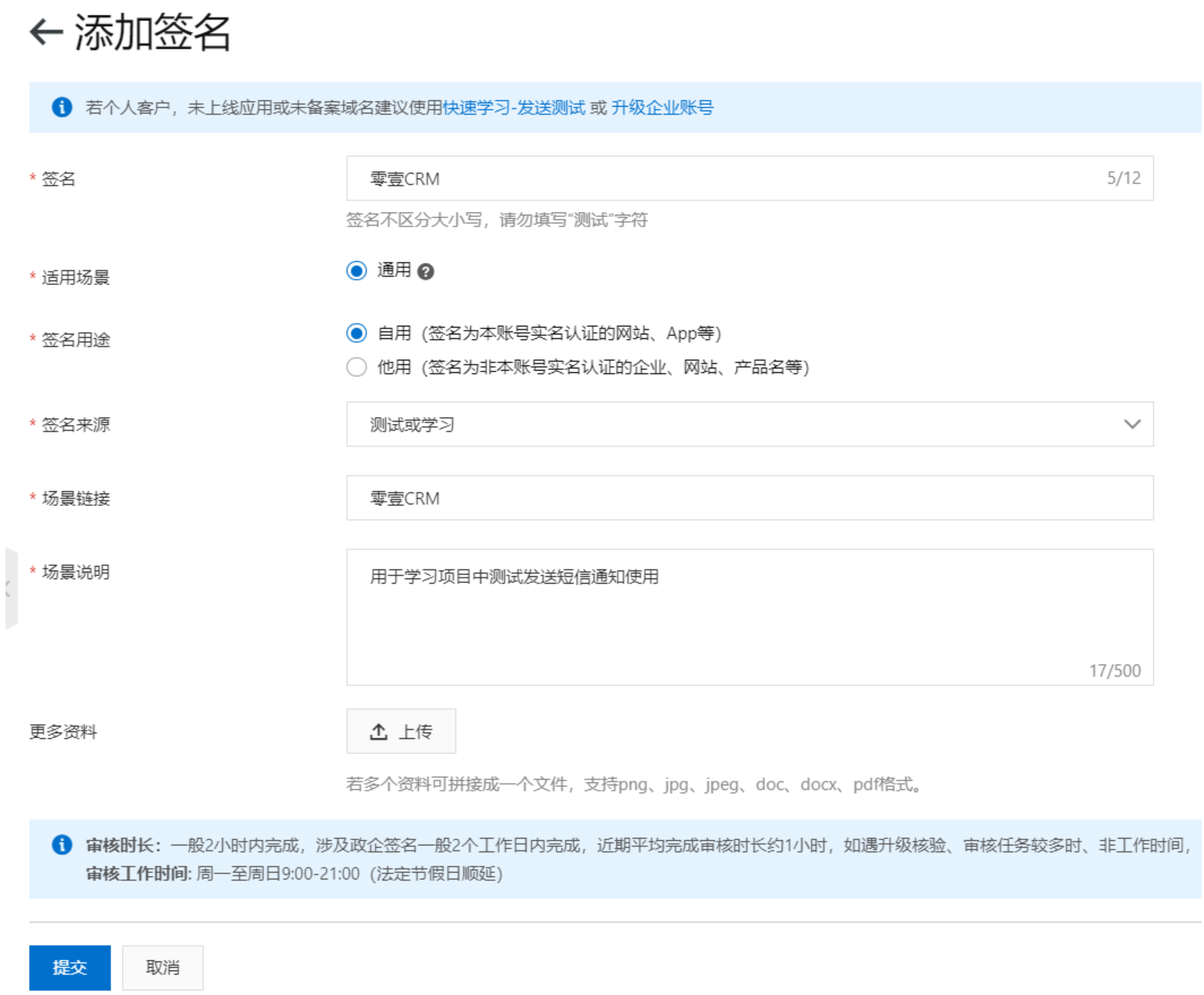
更多功能上线中，欢迎扫码建群

阿里云短信服务严禁发送金融营销、赌博、诈骗、淫秽

1、审核时长一般2小时内完成，涉及政企签名一般

2、签名审核通过须依赖资质审核通过，若资质审

在签名信息页填写对应参数即可，签名页示意图如下


← 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

* 签名：零壹CRM 5/12
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）

* 签名来源：测试或学习

* 场景链接：零壹CRM

* 场景说明：用于学习项目中测试发送短信通知使用 17/500

更多资料：上传
若多个资料可拼接成一个文件，支持png、jpg、jpeg、doc、docx、pdf格式。

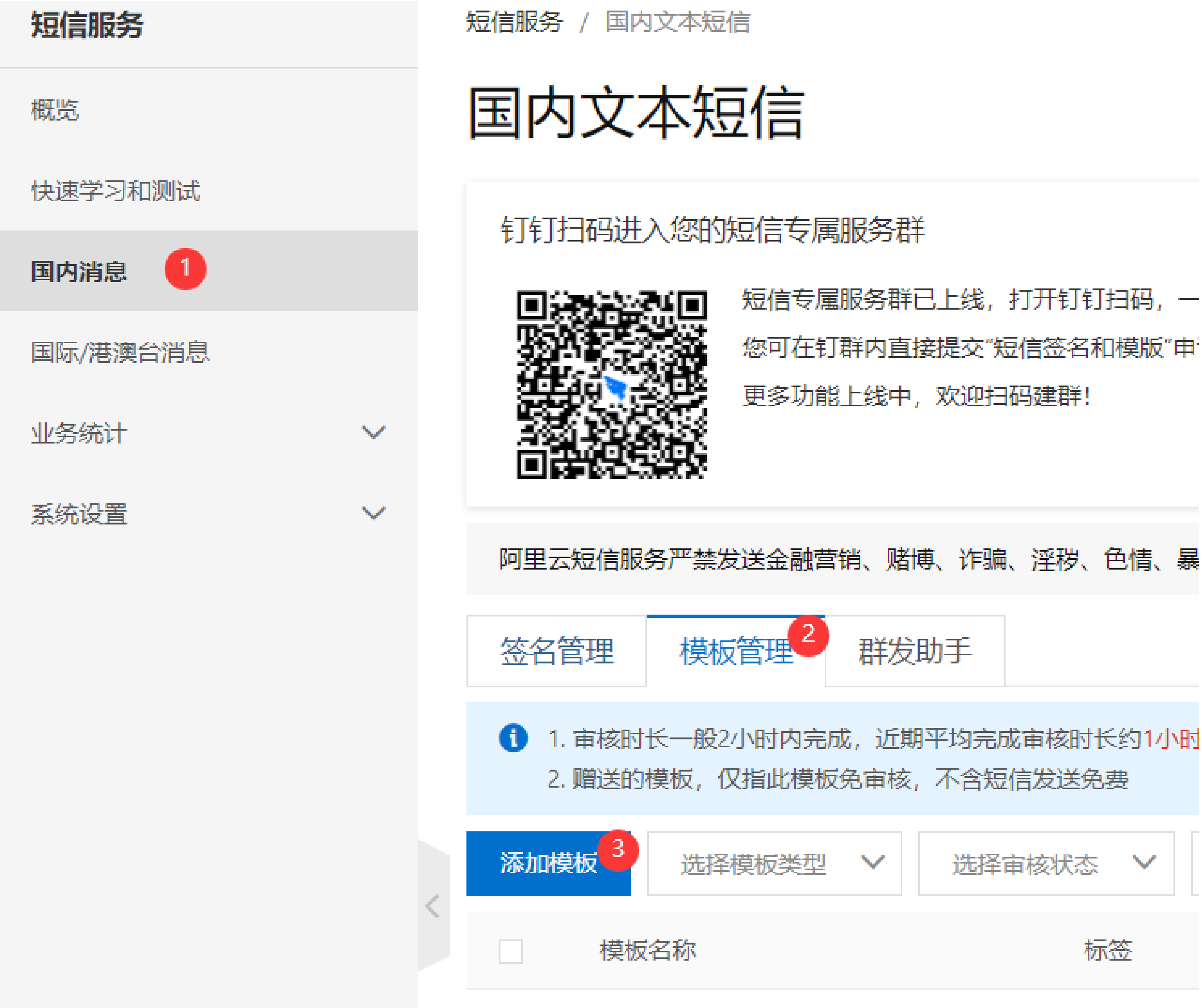
审核时长：一般2小时内完成，涉及政企签名一般2个工作日内完成，近期平均完成审核时长约1小时，如遇升级核验、审核任务较多时、非工作时间，
审核工作时间: 周一至周日9:00-21:00（法定节假日顺延）

注意事项

- 对于个人认证用户，短信签名申请内容必须为已上线App名称、已在工信部备案的网站名称等，暂不支持个人用户申请未上线的业务。
- 对于企业认证用户，如果网站、App、小程序或公众号尚未上线，建议申请企业简称为签名。待网站、App、小程序或公众号上线之后，请及时修改该签名，并重新提交审核。
- 审核中的短信签名不支持撤回。
- 您可以使用默认签名"阿里云IDaaS"和默认模板"快捷登录验证码短信"。如果需要使用自定义的签名和模板，可自行添加，完成审核后即可使用。

6 创建正文模板

在短信服务控制台中找到模板管理，如下图所示



钉钉扫码进入您的短信专属服务群

短信专属服务群已上线，打开钉钉扫码，一

您可在钉群内直接提交"短信签名和模版"申

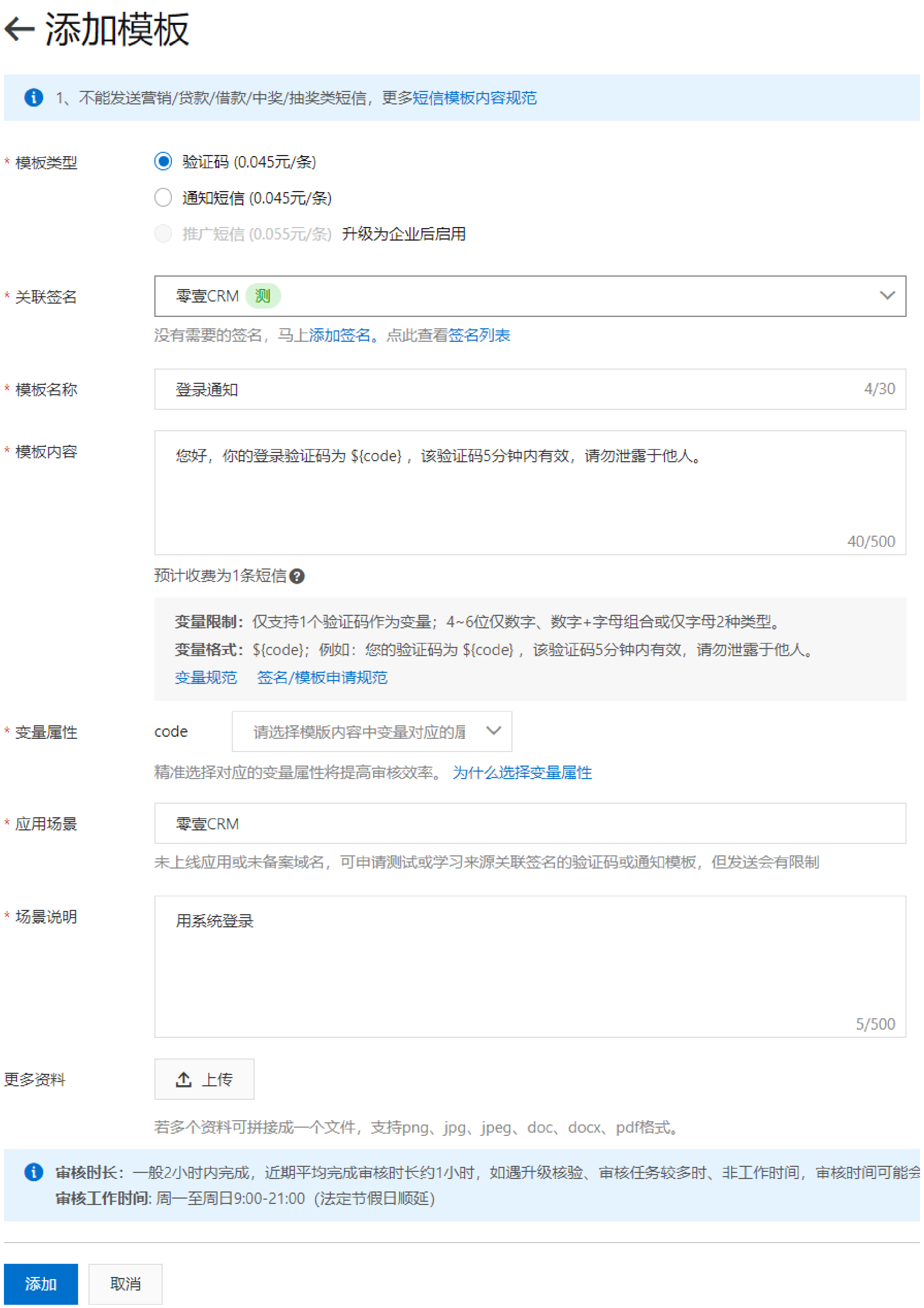
更多功能上线中，欢迎扫码建群！

阿里云短信服务严禁发送金融营销、赌博、诈骗、淫秽、色情、暴

1. 审核时长一般2小时内完成，近期平均完成审核时长约1小时

2. 赠送的模板，仅指此模板免审核，不含短信发送免费

在模板信息页填写对应参数即可，模板页示意图如下



← 添加模板

ℹ 1、不能发送营销/贷款/借款/中奖/抽奖类短信，更多短信模板内容规范

* 模板类型
● 验证码 (0.045元/条)
○ 通知短信 (0.045元/条)
○ 推广短信 (0.055元/条) 升级为企业后启用

* 关联签名：零壹CRM 测
没有需要的签名，马上添加签名。点此查看签名列表

* 模板名称：登录通知 4/30

* 模板内容：您好，你的登录验证码为 ${code}，该验证码5分钟内有效，请勿泄露于他人。 40/500

预计收费为1条短信 ?

变量限制：仅支持1个验证码作为变量；4~6位仅数字、数字+字母组合或仅字母2种类型。
变量格式：${code}；例如：您的验证码为 ${code}，该验证码5分钟内有效，请勿泄露于他人。
变量规范 签名/模板申请规范

* 变量属性：code 请选择模版内容中变量对应的屌
精准选择对应的变量属性将提高审核效率。 为什么选择变量属性

* 应用场景：零壹CRM
未上线应用或未备案域名，可申请测试或学习来源关联签名的验证码或通知模板，但发送会有限制

* 场景说明：用系统登录 5/500

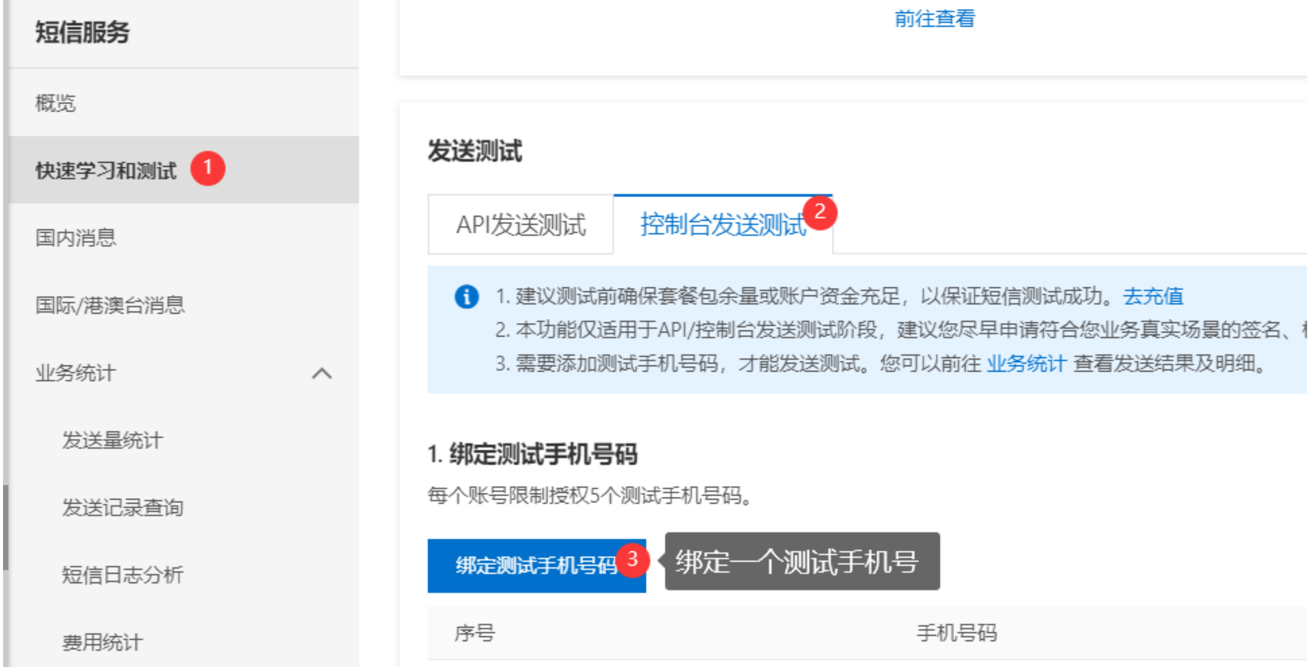
更多资料 上传
若多个资料可拼接成一个文件，支持png、jpg、jpeg、doc、docx、pdf格式。

ℹ **审核时长：**一般2小时内完成，近期平均完成审核时长约1小时，如遇升级核验、审核任务较多时、非工作时间，审核时间可能会
审核工作时间：周一至周日9:00-21:00（法定节假日顺延）

7 测试发送短信

下面我们在控制台测试发送一下短信

首先绑定一个测试手机号



发送测试

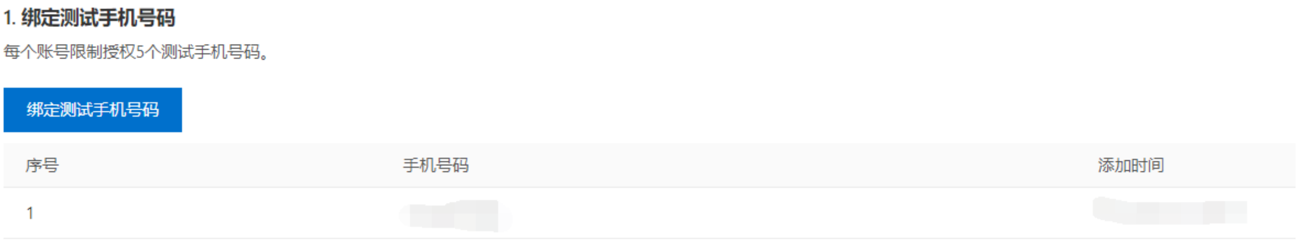
1. 建议测试前确保套餐包余量或账户资金充足，以保证短信测试成功。去充值
2. 本功能仅适用于API/控制台发送测试阶段，建议您尽早申请符合您业务真实场景的签名、t
3. 需要添加测试手机号码，才能发送测试。您可以前往 业务统计 查看发送结果及明细。

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。

序号	手机号码

绑定成功后可以在测试列表中看到测试手机号

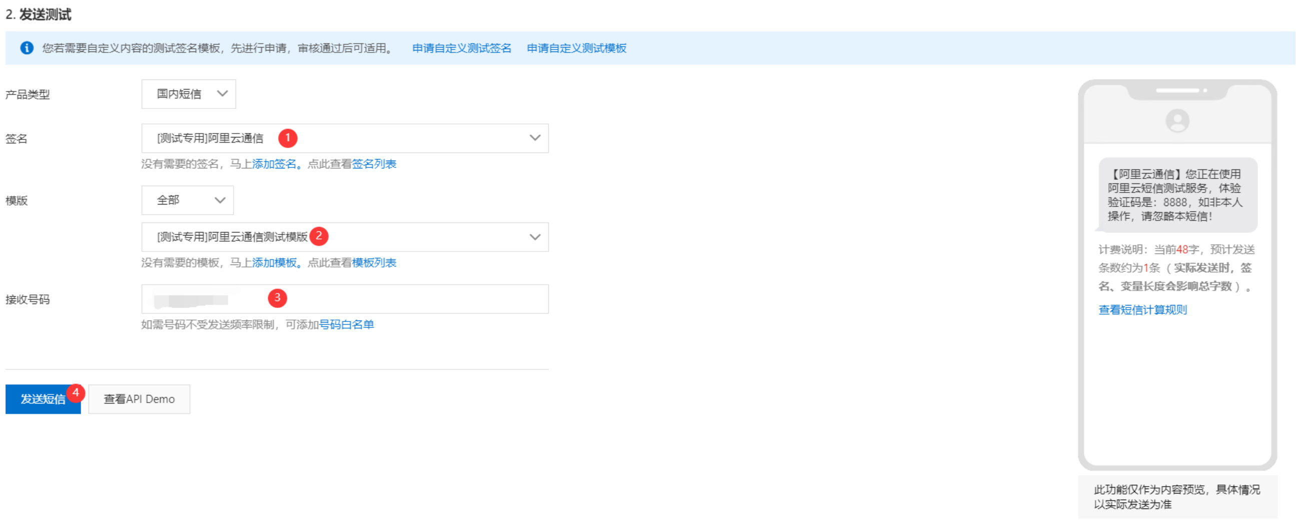


1. 绑定测试手机号码

每个账号限制授权5个测试手机号码。

序号	手机号码	添加时间
1	[redacted]	[redacted]

发送短信



2. 发送测试

您若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	[测试专用]阿里云通信 1 没有需要的签名，马上添加签名。点此查看签名列表
模版	全部 [测试专用]阿里云通信测试模版 2 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	[redacted] 3 如需号码不受发送频率限制，可添加号码白名单

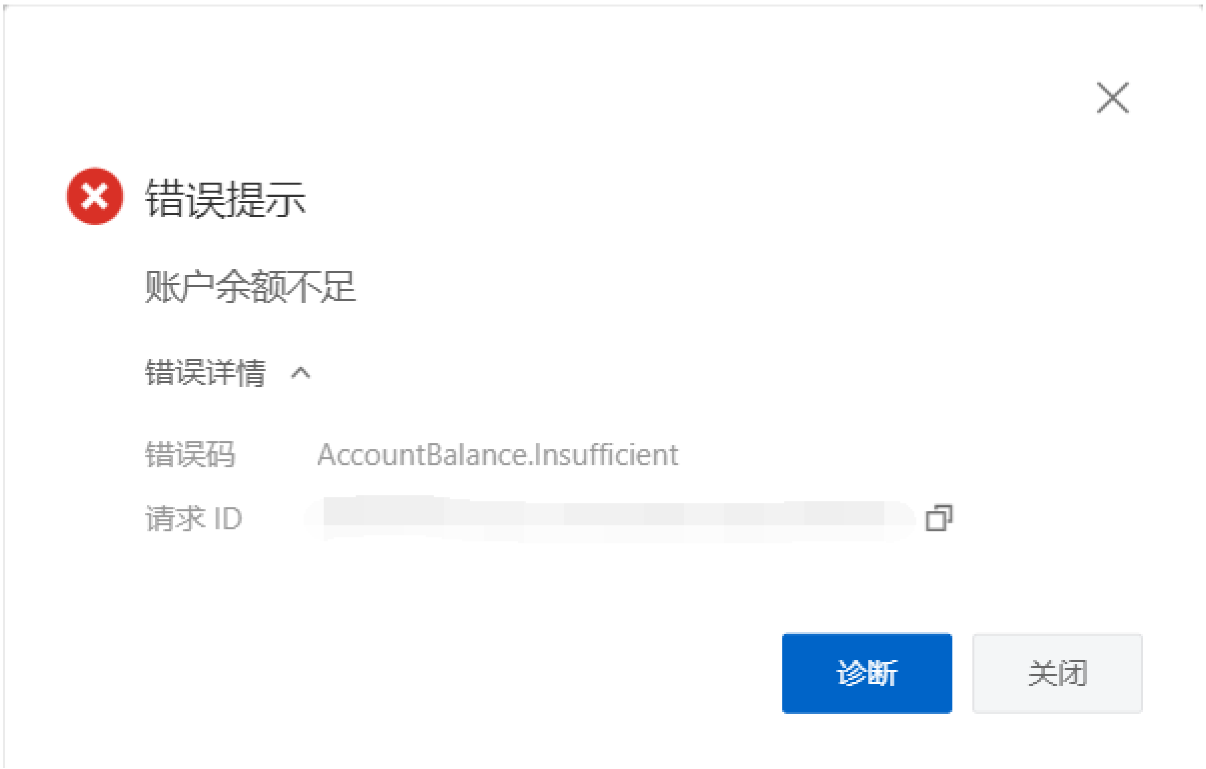
【阿里云通信】您正在使用阿里云短信测试服务，体验验证码是：8888，如非本人操作，请忽略本短信！

计费说明：当前48字，预计发送条数约为1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

此功能仅作为内容预览，具体情况以实际发送为准

发送的时候会提示账号余额不足，如下图所示



错误提示

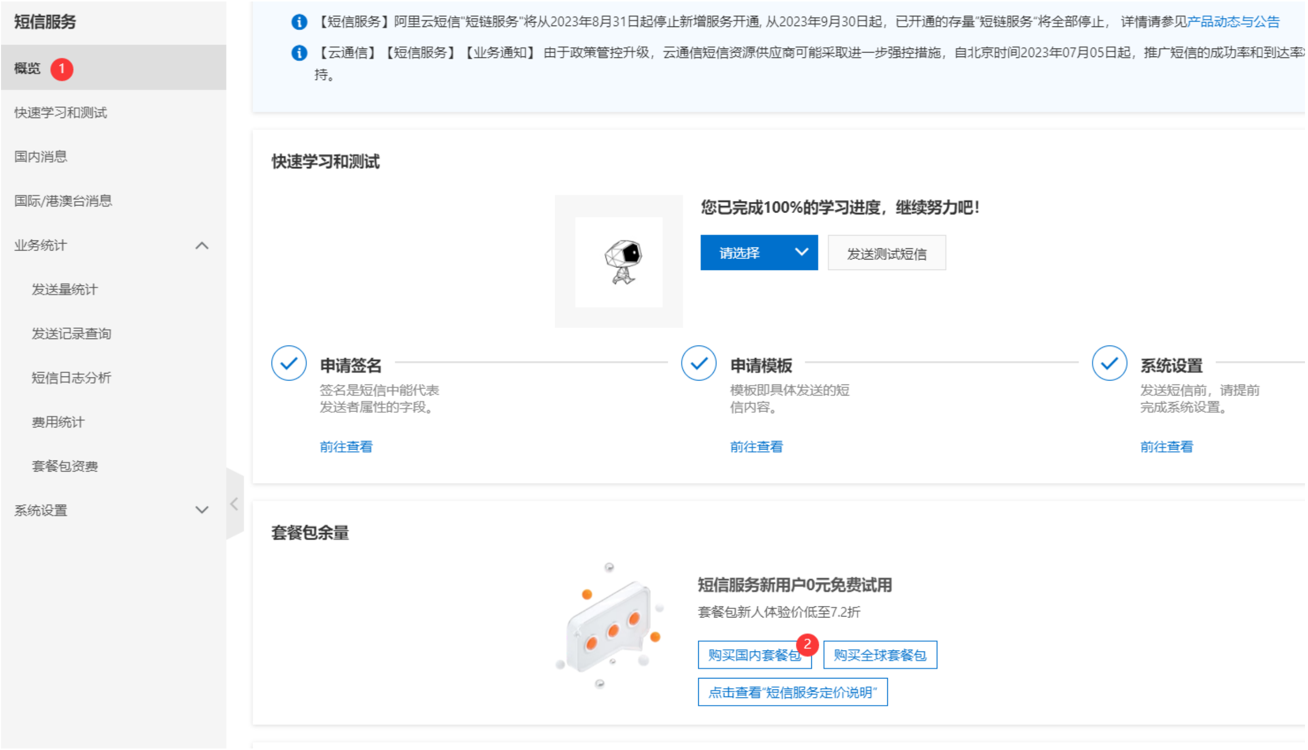
账户余额不足

错误详情

错误码 AccountBalance.Insufficient

请求 ID [redacted]

此时就需要你购买短信套餐包，可以按照下面的步骤去购买套餐包



【短信服务】阿里云短信"短链服务"将从2023年8月31日起停止新增服务开通, 从2023年9月30日起，已开通的存量"短链服务"将全部停止， 详情请参见产品动态与公告

【云通信】【短信服务】【业务通知】 由于政策管控升级，云通信短信资源供应商可能采取进一步强控措施，自北京时间2023年07月05日起，推广短信的成功率和到达率持。

快速学习和测试

您已完成100%的学习进度，继续努力吧!

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

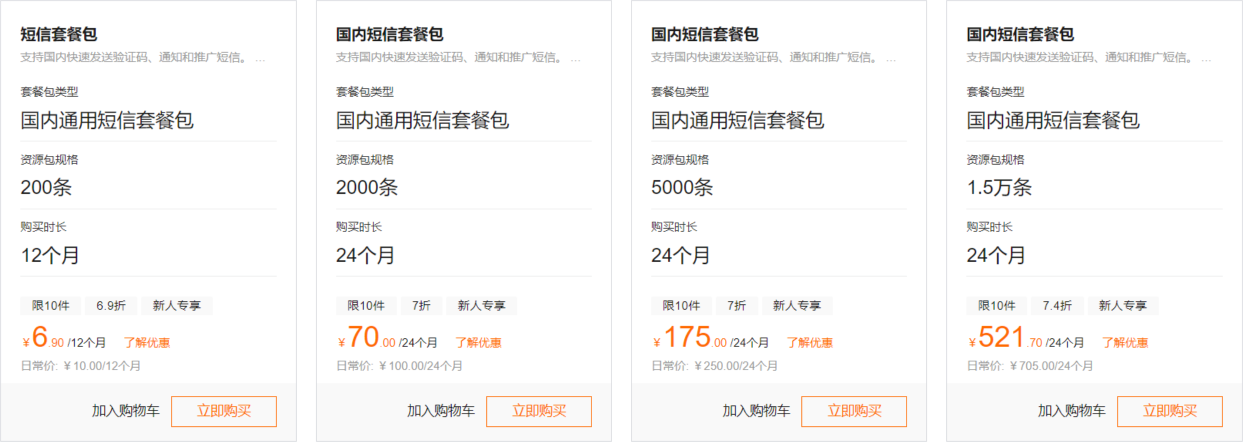
申请模板
模板即具体发送的短信内容。
前往查看

系统设置
发送短信前，请提前完成系统设置。
前往查看

套餐包余量

短信服务新用户0元免费试用
套餐包新人体验价低至7.2折

打开了套餐界面



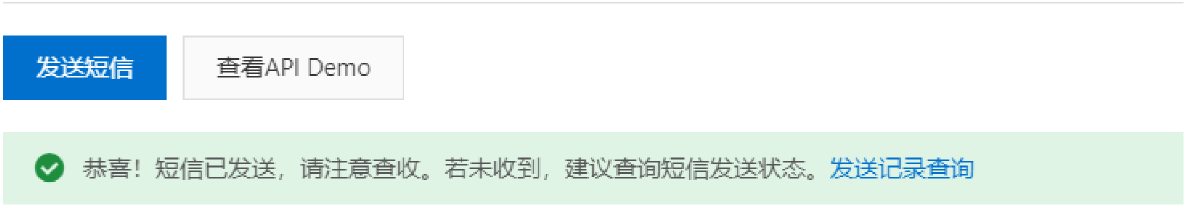
短信套餐包
支持国内快速发送验证码、通知和推广短信。 ...
套餐包类型
国内通用短信套餐包
资源包规格
200条
购买时长
12个月
限10件 6.9折 新人专享
¥6.90 /12个月 了解优惠
日常价: ¥10.00/12个月

国内短信套餐包
支持国内快速发送验证码、通知和推广短信。 ...
套餐包类型
国内通用短信套餐包
资源包规格
2000条
购买时长
24个月
限10件 7折 新人专享
¥70.00 /24个月 了解优惠
日常价: ¥100.00/24个月

国内短信套餐包
支持国内快速发送验证码、通知和推广短信。 ...
套餐包类型
国内通用短信套餐包
资源包规格
5000条
购买时长
24个月
限10件 7折 新人专享
¥175.00 /24个月 了解优惠
日常价: ¥250.00/24个月

国内短信套餐包
支持国内快速发送验证码、通知和推广短信。 ...
套餐包类型
国内通用短信套餐包
资源包规格
1.5万条
购买时长
24个月
限10件 7.4折 新人专享
¥521.70 /24个月 了解优惠
日常价: ¥705.00/24个月

买个最便宜的体验一下就行了，充值成功后，再次发送就能成功，下面是发送成功后的提示效果，同时你发送的手机也能收到消息



恭喜！短信已发送，请注意查收。若未收到，建议查询短信发送状态。发送记录查询

二、项目中使用

帮助文档：https://help.aliyun.com/document_detail/2391766.html?spm=a2c4g.419298.0.0.1cd979beZl7maa

SDK地址：https://help.aliyun.com/document_detail/212725.html?spm=a2c4g.2391766.0.0.4e2e79be97MI0b

Java中使用

https://next.api.aliyun.com/api-tools/sdk/Dysmsapi?version=2017-05-25&language=java-tea&tab=primer-doc

1 添加依赖库

```xml
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>dysmsapi20170525</artifactId>
  <version>2.0.24</version>
</dependency>
```

2 示例代码

- 准备必要参数

```java
Config config = new Config()
    // 您的AccessKey ID
    .setAccessKeyId(accessKeyId)
    // 您的AccessKey Secret
    .setAccessKeySecret(accessKeySecret);
// 访问的域名
config.endpoint = "dysmsapi.aliyuncs.com";
```

- 完整示例代码

```java
// This file is auto-generated, don't edit it. Thanks.
package com.aliyun.sample;

import com.aliyun.tea.*;

public class Sample {

    /**
     * 使用AK&SK初始化账号Client
     * @param accessKeyId
     * @param accessKeySecret
     * @return Client
     * @throws Exception
     */
    public static com.aliyun.dysmsapi20170525.Client createClient(String accessKeyId, String accessKeySecret) throws Exception {
        com.aliyun.teaopenapi.models.Config config = new com.aliyun.teaopenapi.models.Config()
                // 必填，您的 AccessKey ID
                .setAccessKeyId(accessKeyId)
                // 必填，您的 AccessKey Secret
                .setAccessKeySecret(accessKeySecret);
        // Endpoint 请参考 https://api.aliyun.com/product/Dysmsapi
        config.endpoint = "dysmsapi.aliyuncs.com";
        return new com.aliyun.dysmsapi20170525.Client(config);
    }

    public static void main(String[] args_) throws Exception {
        java.util.List<String> args = java.util.Arrays.asList(args_);
        // 请确保代码运行环境设置了环境变量 ALIBABA_CLOUD_ACCESS_KEY_ID 和 ALIBABA_CLOUD_ACCESS_KEY_SECRET。
        // 工程代码泄露可能会导致 AccessKey 泄露，并威胁账号下所有资源的安全性。以下代码示例使用环境变量获取 AccessKey 的方式进行调用，仅供参考，建议使用更安全的 STS 方式，更多鉴权访问方式请参见：https://help.aliyun.com/document_detail/378657.html
        com.aliyun.dysmsapi20170525.Client client = Sample.createClient(System.getenv("ALIBABA_CLOUD_ACCESS_KEY_ID"), System.getenv("ALIBABA_CLOUD_ACCESS_KEY_SECRET"));
        com.aliyun.dysmsapi20170525.models.SendSmsRequest sendSmsRequest = new com.aliyun.dysmsapi20170525.models.SendSmsRequest()
                .setPhoneNumbers("your_value")
                .setSignName("your_value");
        try {
            // 复制代码运行请自行打印 API 的返回值
            client.sendSmsWithOptions(sendSmsRequest, new com.aliyun.teautil.models.RuntimeOptions());
        } catch (TeaException error) {
            // 如有需要，请打印 error
            com.aliyun.teautil.Common.assertAsString(error.message);
        } catch (Exception _error) {
            TeaException error = new TeaException(_error.getMessage(), _error);
            // 如有需要，请打印 error
            com.aliyun.teautil.Common.assertAsString(error.message);
        }
    }
}
```

C++中使用

下面教程主要讲Linux环境下面的使用教程。

SDK 2.0

帮助文档：https://next.api.aliyun.com/api-tools/sdk/Dysmsapi?version=2017-05-25&language=cpp-tea&tab=primer-doc

首先看一下环境要求：

- 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
- 安装依赖库，具体参考必备依赖库。

1 获取源代码

```
mkdir /home/aliyun
cd /home/aliyun
git clone https://github.com/alibabacloud-sdk-cpp/dysmsapi-20170525.git
```

2 安装依赖

- 安装`boost`和`openssl`

```
dnf update -y
dnf install -y boost-devel openssl-devel libcurl-devel
yum install -y perl-IPC-Cmd
```

- 安装`CPPRestSDK`

参考链接：

https://gist.github.com/coin8086/488580e650a552bdfbd0f858c193a26e

https://github.com/Microsoft/cpprestsdk/wiki/How-to-build-for-Linux

```
git clone https://github.com/Microsoft/cpprestsdk.git casablanca
cd casablanca
git submodule update --init
cd Release
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release -DWERROR=OFF
make
#将编译文件安装到：/usr/local/lib64和/usr/local/include目录下面
make install
```

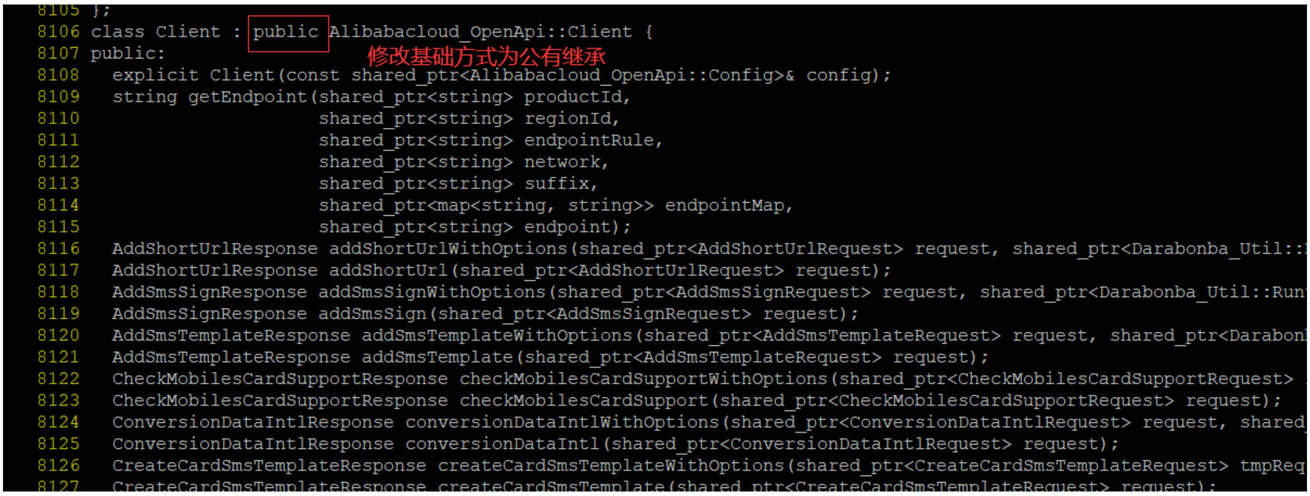
- 安装Jsoncpp

```
git clone https://github.com/open-source-parsers/jsoncpp.git
cd jsoncpp
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=release -DBUILD_STATIC_LIBS=OFF -DBUILD_SHARED_LIBS=ON ..
make
#将编译文件安装到：/usr/local/lib64和/usr/local/include目录下面
make install
```

3 编译阿里云SDK

```
cd /home/aliyun/dysmsapi-20170525
sh scripts/build.sh
#将编译文件安装到：/usr/local/lib64、/usr/local/include、/usr/local/share目录下面
sh scripts/install.sh
```

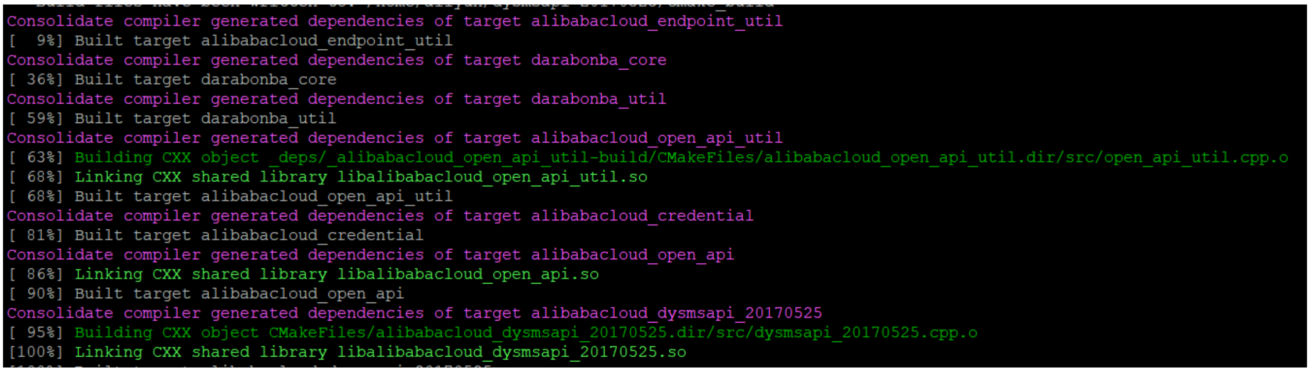
编译之前修改一下对源码（dysmsapi-20170525.hpp）进行修改



```
};
class Client : public Alibabacloud_OpenApi::Client {
public:
  explicit Client(const shared_ptr<Alibabacloud_OpenApi::Config>& config);
  string getEndpoint(shared_ptr<string> productId,
                     shared_ptr<string> regionId,
                     shared_ptr<string> endpointRule,
                     shared_ptr<string> network,
                     shared_ptr<string> suffix,
                     shared_ptr<map<string, string>> endpointMap,
                     shared_ptr<string> endpoint);
  AddShortUrlResponse addShortUrlWithOptions(shared_ptr<AddShortUrlRequest> request, shared_ptr<Darabonba_Util::
  AddShortUrlResponse addShortUrl(shared_ptr<AddShortUrlRequest> request);
  AddSmsSignResponse addSmsSignWithOptions(shared_ptr<AddSmsSignRequest> request, shared_ptr<Darabonba_Util::Run
  AddSmsSignResponse addSmsSign(shared_ptr<AddSmsSignRequest> request);
  AddSmsTemplateResponse addSmsTemplateWithOptions(shared_ptr<AddSmsTemplateRequest> request, shared_ptr<Darabon
  AddSmsTemplateResponse addSmsTemplate(shared_ptr<AddSmsTemplateRequest> request);
  CheckMobilesCardSupportResponse checkMobilesCardSupportWithOptions(shared_ptr<CheckMobilesCardSupportRequest>
  CheckMobilesCardSupportResponse checkMobilesCardSupport(shared_ptr<CheckMobilesCardSupportRequest> request);
  ConversionDataIntlResponse conversionDataIntlWithOptions(shared_ptr<ConversionDataIntlRequest> request, shared
  ConversionDataIntlResponse conversionDataIntl(shared_ptr<ConversionDataIntlRequest> request);
  CreateCardSmsTemplateResponse createCardSmsTemplateWithOptions(shared_ptr<CreateCardSmsTemplateRequest> tmpReq
  CreateCardSmsTemplateResponse createCardSmsTemplate(shared_ptr<CreateCardSmsTemplateRequest> request);
```

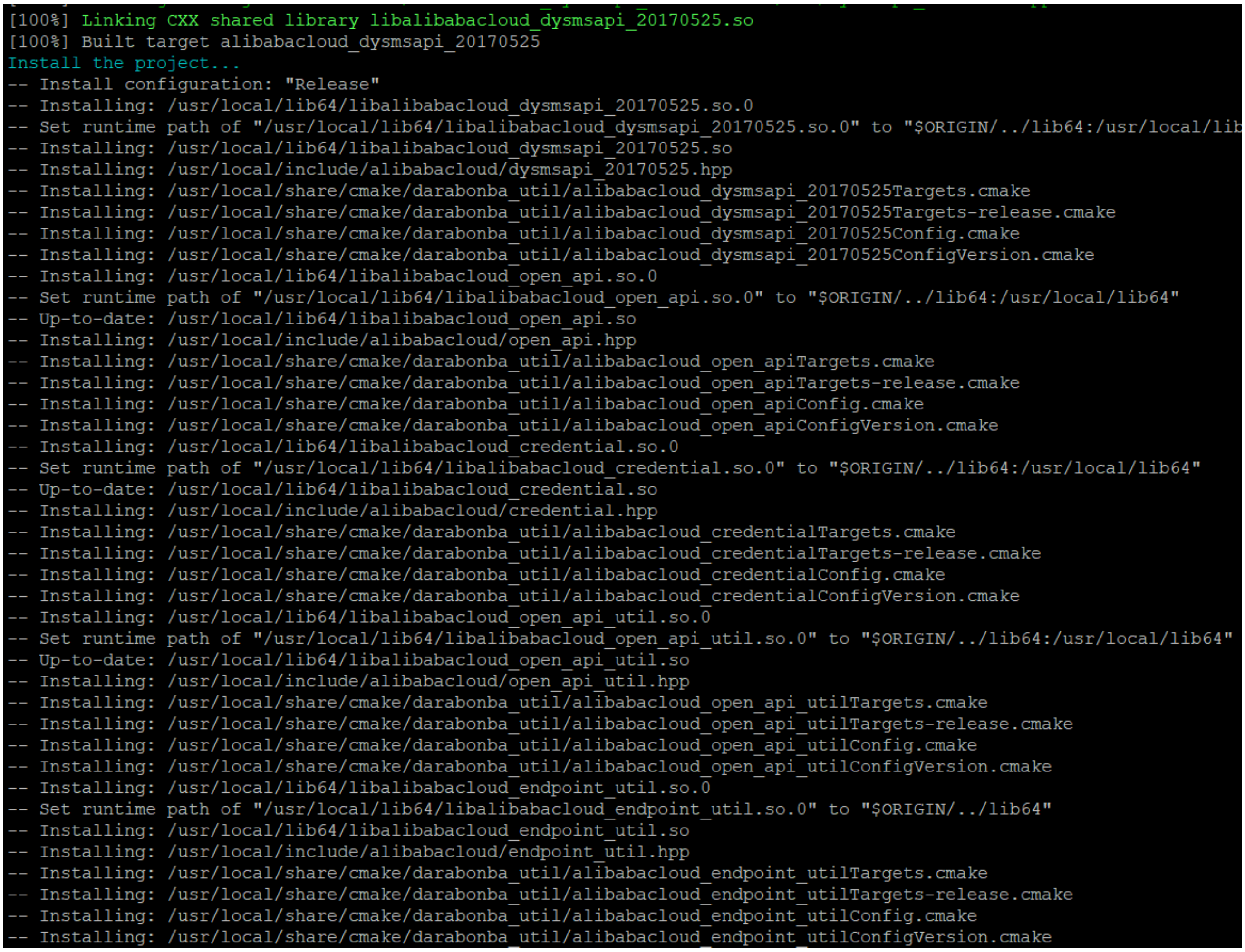
修改基础方式为公有继承

编译通过截图



```
Consolidate compiler generated dependencies of target alibabacloud_endpoint_util
[  9%] Built target alibabacloud_endpoint_util
Consolidate compiler generated dependencies of target darabonba_core
[ 36%] Built target darabonba_core
Consolidate compiler generated dependencies of target darabonba_util
[ 59%] Built target darabonba_util
Consolidate compiler generated dependencies of target alibabacloud_open_api_util
[ 63%] Building CXX object _deps/_alibabacloud_open_api_util-build/CMakeFiles/alibabacloud_open_api_util.dir/src/open_api_util.cpp.o
[ 68%] Linking CXX shared library libalibabacloud_open_api_util.so
[ 68%] Built target alibabacloud_open_api_util
Consolidate compiler generated dependencies of target alibabacloud_credential
[ 81%] Built target alibabacloud_credential
Consolidate compiler generated dependencies of target alibabacloud_open_api
[ 86%] Linking CXX shared library libalibabacloud_open_api.so
[ 90%] Built target alibabacloud_open_api
Consolidate compiler generated dependencies of target alibabacloud_dysmsapi_20170525
[ 95%] Building CXX object CMakeFiles/alibabacloud_dysmsapi_20170525.dir/src/dysmsapi_20170525.cpp.o
[100%] Linking CXX shared library libalibabacloud_dysmsapi_20170525.so
```

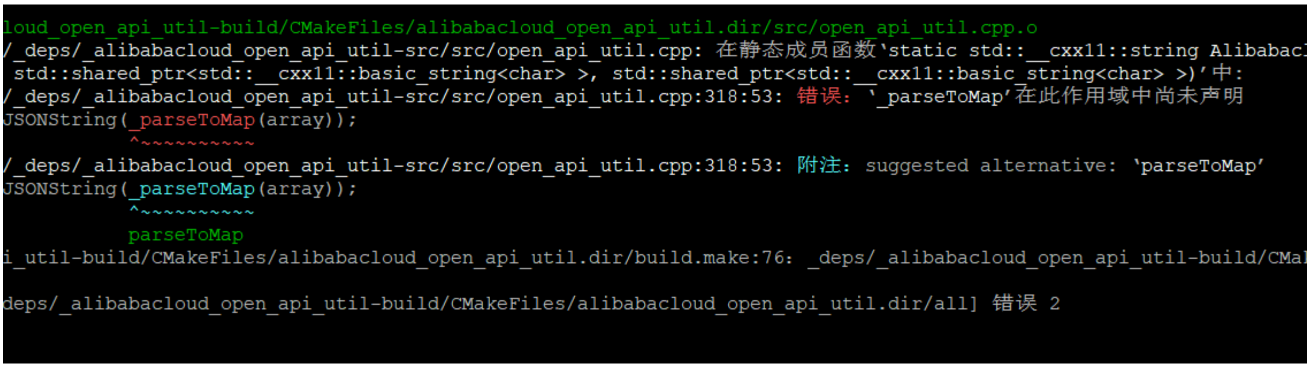
安装完成截图



```
[100%] Linking CXX shared library libalibabacloud_dysmsapi_20170525.so
[100%] Built target alibabacloud_dysmsapi_20170525
Install the project...
-- Install configuration: "Release"
-- Installing: /usr/local/lib64/libalibabacloud_dysmsapi_20170525.so.0
-- Set runtime path of "/usr/local/lib64/libalibabacloud_dysmsapi_20170525.so.0" to "$ORIGIN/../lib64:/usr/local/lib
-- Installing: /usr/local/lib64/libalibabacloud_dysmsapi_20170525.so
-- Installing: /usr/local/include/alibabacloud/dysmsapi_20170525.hpp
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_dysmsapi_20170525Targets.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_dysmsapi_20170525Targets-release.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_dysmsapi_20170525Config.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_dysmsapi_20170525ConfigVersion.cmake
-- Installing: /usr/local/lib64/libalibabacloud_open_api.so.0
-- Set runtime path of "/usr/local/lib64/libalibabacloud_open_api.so.0" to "$ORIGIN/../lib64:/usr/local/lib64"
-- Up-to-date: /usr/local/lib64/libalibabacloud_open_api.so
-- Installing: /usr/local/include/alibabacloud/open_api.hpp
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_apiTargets.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_apiTargets-release.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_apiConfig.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_apiConfigVersion.cmake
-- Installing: /usr/local/lib64/libalibabacloud_credential.so.0
-- Set runtime path of "/usr/local/lib64/libalibabacloud_credential.so.0" to "$ORIGIN/../lib64:/usr/local/lib64"
-- Up-to-date: /usr/local/lib64/libalibabacloud_credential.so
-- Installing: /usr/local/include/alibabacloud/credential.hpp
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_credentialTargets.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_credentialTargets-release.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_credentialConfig.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_credentialConfigVersion.cmake
-- Installing: /usr/local/lib64/libalibabacloud_open_api_util.so.0
-- Set runtime path of "/usr/local/lib64/libalibabacloud_open_api_util.so.0" to "$ORIGIN/../lib64:/usr/local/lib64"
-- Up-to-date: /usr/local/lib64/libalibabacloud_open_api_util.so
-- Installing: /usr/local/include/alibabacloud/open_api_util.hpp
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_api_utilTargets.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_api_utilTargets-release.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_api_utilConfig.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_open_api_utilConfigVersion.cmake
-- Installing: /usr/local/lib64/libalibabacloud_endpoint_util.so.0
-- Set runtime path of "/usr/local/lib64/libalibabacloud_endpoint_util.so.0" to "$ORIGIN/../lib64"
-- Installing: /usr/local/lib64/libalibabacloud_endpoint_util.so
-- Installing: /usr/local/include/alibabacloud/endpoint_util.hpp
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_endpoint_utilTargets.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_endpoint_utilTargets-release.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_endpoint_utilConfig.cmake
-- Installing: /usr/local/share/cmake/darabonba_util/alibabacloud_endpoint_utilConfigVersion.cmake
```

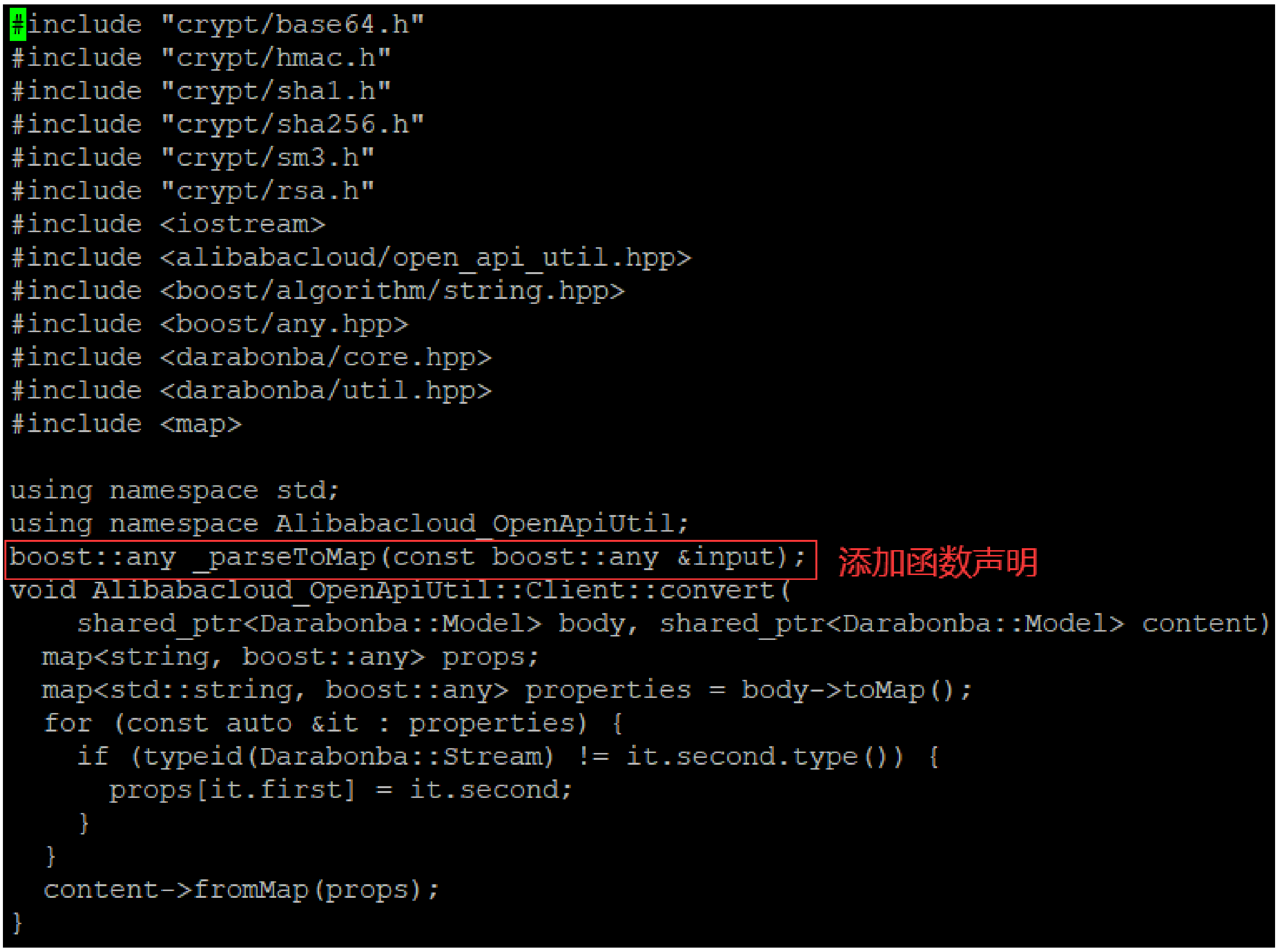
编译过程可能会遇到的错误

错误1：



```
loud_open_api_util-build/CMakeFiles/alibabacloud_open_api_util.dir/src/open_api_util.cpp.o
/_deps/_alibabacloud_open_api_util-src/src/open_api_util.cpp: 在静态成员函数‘static std::__cxx11::string Alibabac
 std::shared_ptr<std::__cxx11::basic_string<char> >, std::shared_ptr<std::__cxx11::basic_string<char> >)’中:
/_deps/_alibabacloud_open_api_util-src/src/open_api_util.cpp:318:53: 错误: ‘_parseToMap’在此作用域中尚未声明
JSONString(_parseToMap(array));
           ^~~~~~~~~~~
/_deps/_alibabacloud_open_api_util-src/src/open_api_util.cpp:318:53: 附注: suggested alternative: ‘parseToMap’
JSONString(_parseToMap(array));
           ^~~~~~~~~~~
           parseToMap
i_util-build/CMakeFiles/alibabacloud_open_api_util.dir/build.make:76: _deps/_alibabacloud_open_api_util-build/CMa

deps/_alibabacloud_open_api_util-build/CMakeFiles/alibabacloud_open_api_util.dir/all] 错误 2
```

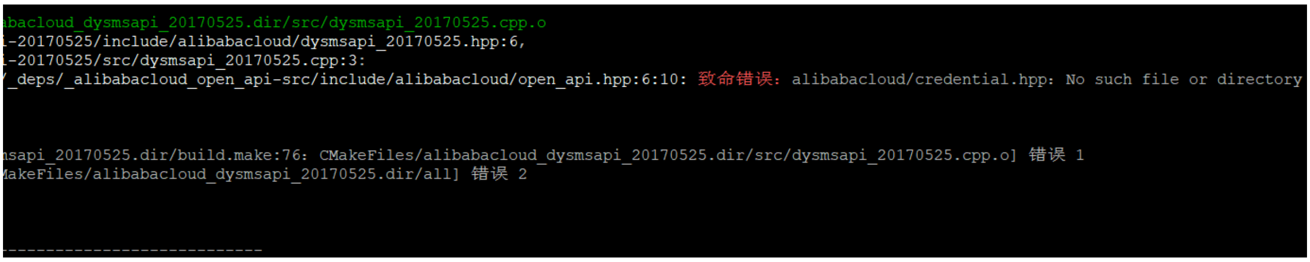
修改对应`open_api_util.cpp`文件，加入函数声明，修改示意如下图所示



```
#include "crypt/base64.h"
#include "crypt/hmac.h"
#include "crypt/sha1.h"
#include "crypt/sha256.h"
#include "crypt/sm3.h"
#include "crypt/rsa.h"
#include <iostream>
#include <alibabacloud/open_api_util.hpp>
#include <boost/algorithm/string.hpp>
#include <boost/any.hpp>
#include <darabonba/core.hpp>
#include <darabonba/util.hpp>
#include <map>

using namespace std;
using namespace Alibabacloud_OpenApiUtil;
boost::any _parseToMap(const boost::any &input);
void Alibabacloud_OpenApiUtil::Client::convert(
    shared_ptr<Darabonba::Model> body, shared_ptr<Darabonba::Model> content)
  map<string, boost::any> props;
  map<std::string, boost::any> properties = body->toMap();
  for (const auto &it : properties) {
    if (typeid(Darabonba::Stream) != it.second.type()) {
      props[it.first] = it.second;
    }
  }
  content->fromMap(props);
}
```

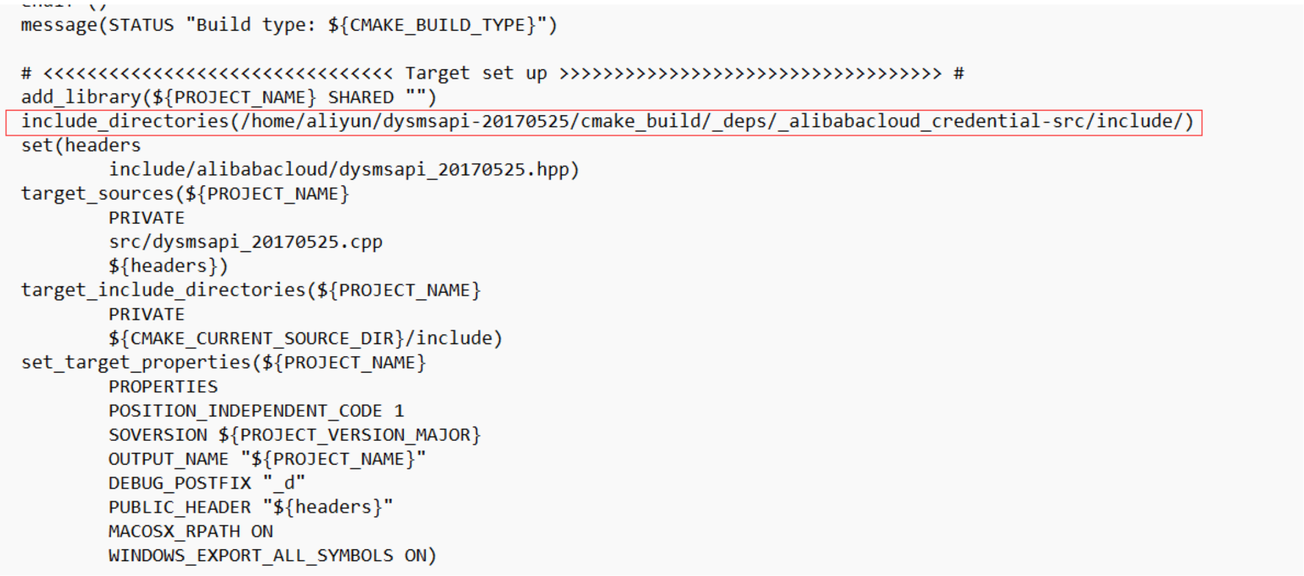
错误2：

如果你编译的是dara-openapi这个库不会出现此错误



```
abacloud_dysmsapi_20170525.dir/src/dysmsapi_20170525.cpp.o
i-20170525/include/alibabacloud/dysmsapi_20170525.hpp:6,
i-20170525/src/dysmsapi_20170525.cpp:3:
/_deps/_alibabacloud_open_api-src/include/alibabacloud/open_api.hpp:6:10: 致命错误：alibabacloud/credential.hpp: No such file or directory


nsapi_20170525.dir/build.make:76: CMakeFiles/alibabacloud_dysmsapi_20170525.dir/src/dysmsapi_20170525.cpp.o] 错误 1
MakeFiles/alibabacloud_dysmsapi_20170525.dir/all] 错误 2



---------------------------
```

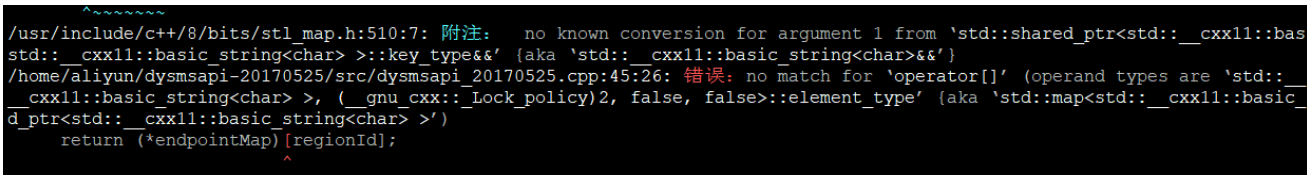
这个问题，修改一下/home/aliyun/dysmsapi-20170525/CMakeLists.txt，补充一下搜索路径即可，参考如下图：



```
endif ()
message(STATUS "Build type: ${CMAKE_BUILD_TYPE}")

# <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< Target set up >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> #
add_library(${PROJECT_NAME} SHARED "")
include_directories(/home/aliyun/dysmsapi-20170525/cmake_build/_deps/_alibabacloud_credential-src/include/)
set(headers
        include/alibabacloud/dysmsapi_20170525.hpp)
target_sources(${PROJECT_NAME}
        PRIVATE
        src/dysmsapi_20170525.cpp
        ${headers})
target_include_directories(${PROJECT_NAME}
        PRIVATE
        ${CMAKE_CURRENT_SOURCE_DIR}/include)
set_target_properties(${PROJECT_NAME}
        PROPERTIES
        POSITION_INDEPENDENT_CODE 1
        SOVERSION ${PROJECT_VERSION_MAJOR}
        OUTPUT_NAME "${PROJECT_NAME}"
        DEBUG_POSTFIX "_d"
        PUBLIC_HEADER "${headers}"
        MACOSX_RPATH ON
        WINDOWS_EXPORT_ALL_SYMBOLS ON)
```

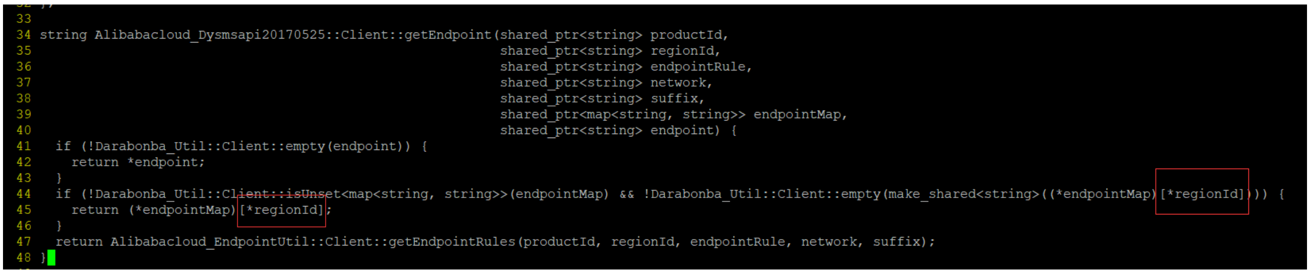
错误3：

如果你编译的是dara-openapi这个库不会出现此错误



```
       ^~~~~~~~
/usr/include/c++/8/bits/stl_map.h:510:7: 附注:   no known conversion for argument 1 from ‘std::shared_ptr<std::__cxx11::bas
std::__cxx11::basic_string<char> >::key_type&&’ {aka ‘std::__cxx11::basic_string<char>&&’}
/home/aliyun/dysmsapi-20170525/src/dysmsapi_20170525.cpp:45:26: 错误: no match for ‘operator[]’ (operand types are ‘std::__
__cxx11::basic_string<char> >, (__gnu_cxx::_Lock_policy)2, false, false>::element_type’ {aka ‘std::map<std::__cxx11::basic_
d_ptr<std::__cxx11::basic_string<char> >’)
     return (*endpointMap)[regionId];
                          ^
```

修改dysmsapi_20170525.cpp，报错位置代码，修改示意如下图所示



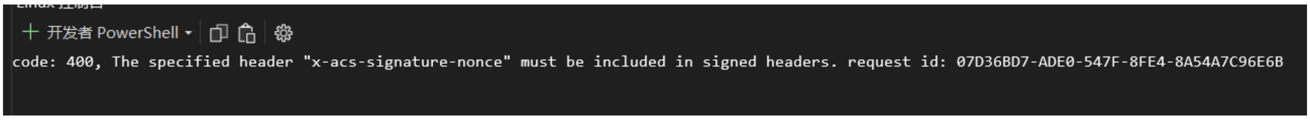
```

string Alibabacloud_Dysmsapi20170525::Client::getEndpoint(shared_ptr<string> productId,
                                                           shared_ptr<string> regionId,
                                                           shared_ptr<string> endpointRule,
                                                           shared_ptr<string> network,
                                                           shared_ptr<string> suffix,
                                                           shared_ptr<map<string, string>> endpointMap,
                                                           shared_ptr<string> endpoint) {
  if (!Darabonba_Util::Client::empty(endpoint)) {
    return *endpoint;
  }
  if (!Darabonba_Util::Client::isUnset<map<string, string>>(endpointMap) && !Darabonba_Util::Client::empty(make_shared<string>((*endpointMap)[*regionId]))) {
    return (*endpointMap)[*regionId];
  }
  return Alibabacloud_EndpointUtil::Client::getEndpointRules(productId, regionId, endpointRule, network, suffix);
}
```

4 示例代码

```
#include <iostream>
#include <alibabacloud/dysmsapi_20170525.hpp>
using namespace std;
int main()
{
    // 创建配置对象
    auto config = std::make_shared<Alibabacloud_OpenApi::Config>();
    config->accessKeyId = std::make_shared<std::string>("ak");
    config->accessKeySecret = std::make_shared<std::string>("sk");
    config->endpoint = std::make_shared<std::string>("dysmsapi.aliyuncs.com");
    // 创建客户端
    Alibabacloud_Dysmsapi20170525::Client client(config);
    // 创建消息请求
    auto req = std::make_shared<Alibabacloud_Dysmsapi20170525::SendSmsRequest>();
    req->phoneNumbers = std::make_shared<std::string>("电话号码");
    req->signName = std::make_shared<std::string>(u8"阿里云短信测试");
    req->templateCode = std::make_shared<std::string>("SMS_154950909");
    req->templateParam = std::make_shared<std::string>("{\"code\":\"996007\"}");
```

```cpp
    try {
        auto resp = client.sendSmsWithOptions(req, std::make_shared<Darabonba_Util::RuntimeOptions>());
        std::cout
            << *resp.body->code << std::endl
            << *resp.body->message << std::endl
            << *resp.body->bizId << std::endl
            << *resp.body->requestId << std::endl;
    }
    catch (const std::exception& e)
    {
        std::cout << e.what() << std::endl;
    }
    return 0;
}
```

提示：编译是通过了但是在发送短信的时候会提示如下错误



```
code: 400, The specified header "x-acs-signature-nonce" must be included in signed headers. request id: 07D36BD7-ADE0-547F-8FE4-8A54A7C96E6B
```

通过询问官方，说是2.0版的SDK暂时无法使用，建议使用其他版本，暂时不使用2.0版本，等待官方修复后再说吧！！

SDK 1.0

2.0使用不了我们就来使用1.0吧。

帮助文档：https://next.api.aliyun.com/api-tools/sdk/Dysmsapi?version=2017-05-25&language=cpp

首先看一下环境要求：

- 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

1 获取源码

```bash
cd /home/aliyun
git clone --branch 1.36.1767 https://github.com/aliyun/aliyun-openapi-cpp-sdk.git
```

2 安装依赖

```bash
dnf install -y libcurl-devel openssl-devel libuuid-devel
yum install -y perl-IPC-Cmd
# 除此之外还需要安装jsoncpp依赖，安装方式参考SDK2.0中的安装方式安装
```

3 编译阿里云SDK

```
cd /home/aliyun/aliyun-openapi-cpp-sdk/
rm -rf sdk_build
mkdir sdk_build
cd sdk_build
cmake -DBUILD_PRODUCT="dysmsapi" -DBUILD_FUNCTION_TESTS=OFF -DBUILD_UNIT_TESTS=OFF -DENABLE_COVERAGE=OFF -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX=/usr/local ..
make
make install
```

4 示例代码

```cpp
#include <iostream>
#include <alibabacloud/core/AlibabaCloud.h>
#include <alibabacloud/core/ClientConfiguration.h>
#include <alibabacloud/dysmsapi/DysmsapiClient.h>
using namespace std;
using namespace AlibabaCloud;
using namespace AlibabaCloud::Dysmsapi;
int main() {
    /* 初始化 SDK */
    AlibabaCloud::InitializeSdk();
    /* 配置实例 */
    ClientConfiguration configuration("cn-hangzhou");
    DysmsapiClient client("ak", "sk", configuration);
    /* 创建API请求并设置参数 */
    Model::SendSmsRequest request;
    request.setPhoneNumbers("电话号码"); // 电话号码
    request.setSignName("阿里云短信测试");// 签名
    request.setTemplateCode("SMS_154950909"); // 模板编码
    request.setTemplateParam("{\"code\":\"996007\"}"); // 模板参数
    /* 发送短信 */
    auto outcome = client.sendSms(request);
    /* 发送异常 */
    if (!outcome.isSuccess())
    {
        /* 异常处理 */
        std::cout << outcome.error().errorCode() << std::endl;
        /* 关闭 SDK */
        AlibabaCloud::ShutdownSdk();
        return -1;
    }

    /* 打印发送结果 */
    std::cout
        << "code:  " << outcome.result().getCode() << std::endl
        << "msg:   " << outcome.result().getMessage() << std::endl
        << "bizid: " << outcome.result().getBizId() << std::endl
        << "reqid: " << outcome.result().requestId() << std::endl;
    /* 关闭 SDK */
    AlibabaCloud::ShutdownSdk();
    return 0;
}
```