

阿里云短信服务入门

Step1：注册阿里云账号

- 注册账号相对简单，按照官方流程傻瓜式注册就行。
- 进行实名认证，建议使用支付宝实名认证，方便快捷，也是官方推荐的实名认证方式！

Step2：申请签名

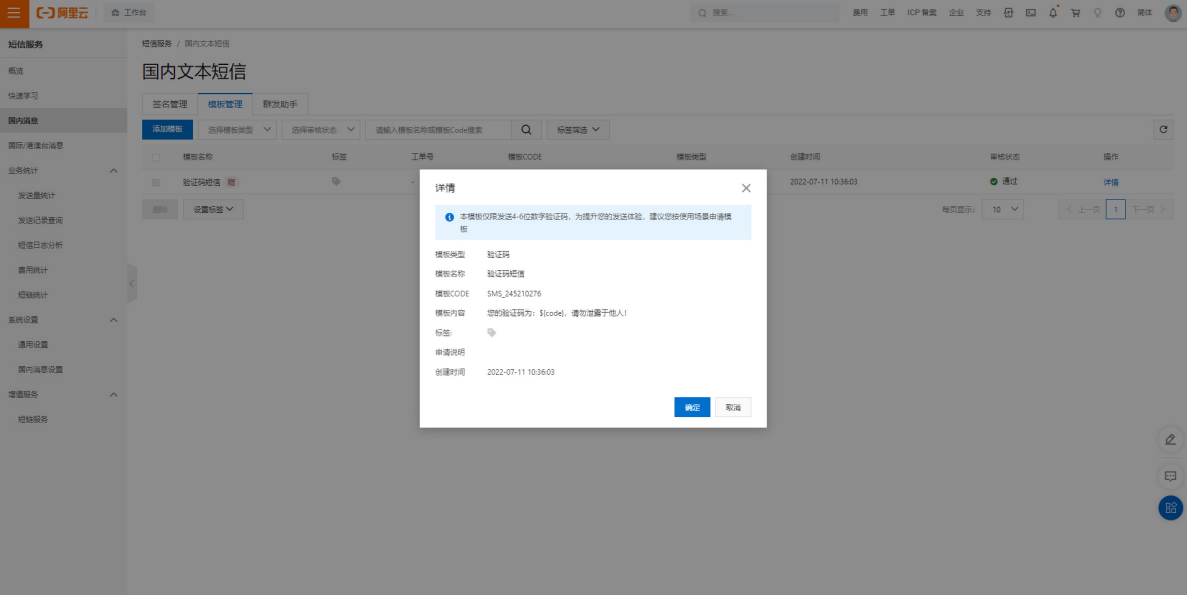
- 选择 国内消息 按钮，申请签名



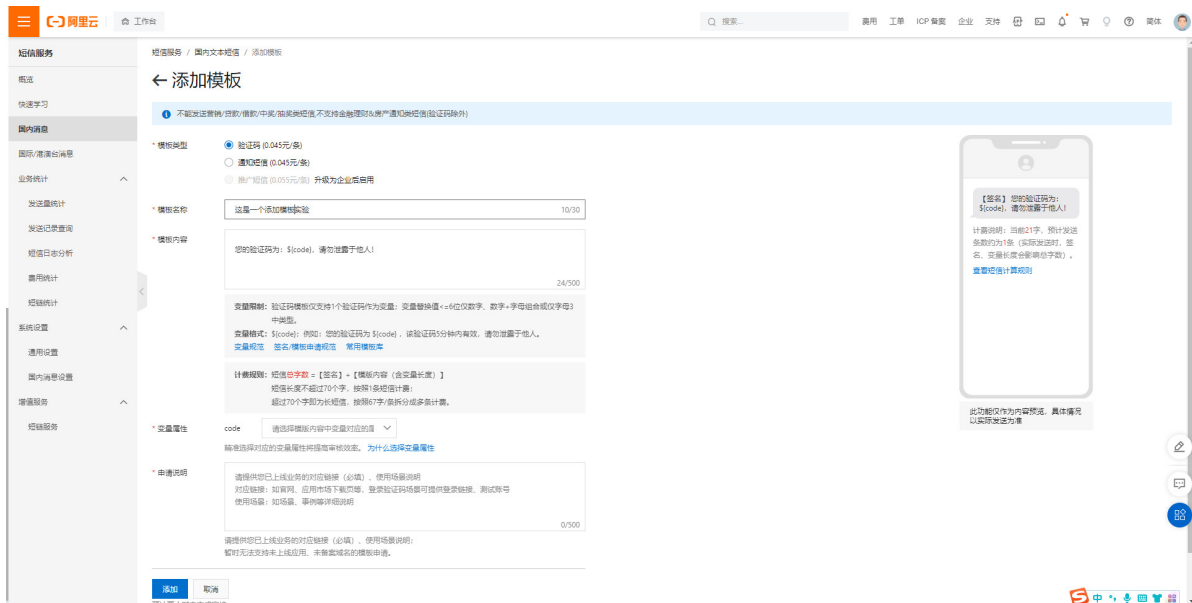
Note: 现在个人比较难申请，比较麻烦，需要上传人多认证资料，这个大家不一定有！

Step3：创建模板

- 官方赠送模板



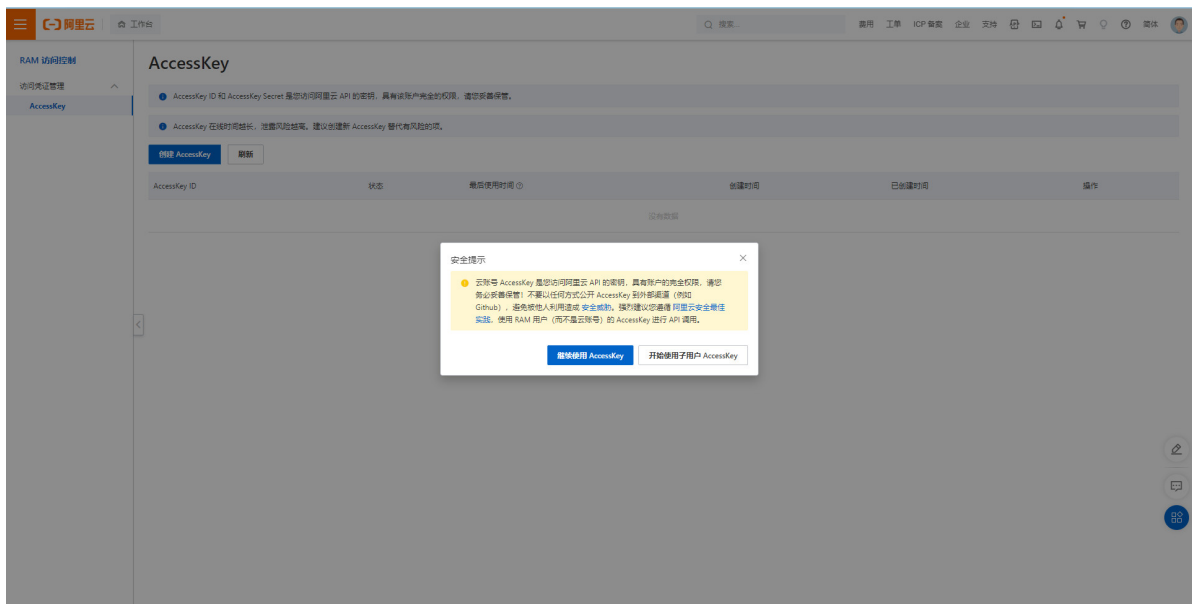
- 创建个人模板---> 点击 添加模板 按钮



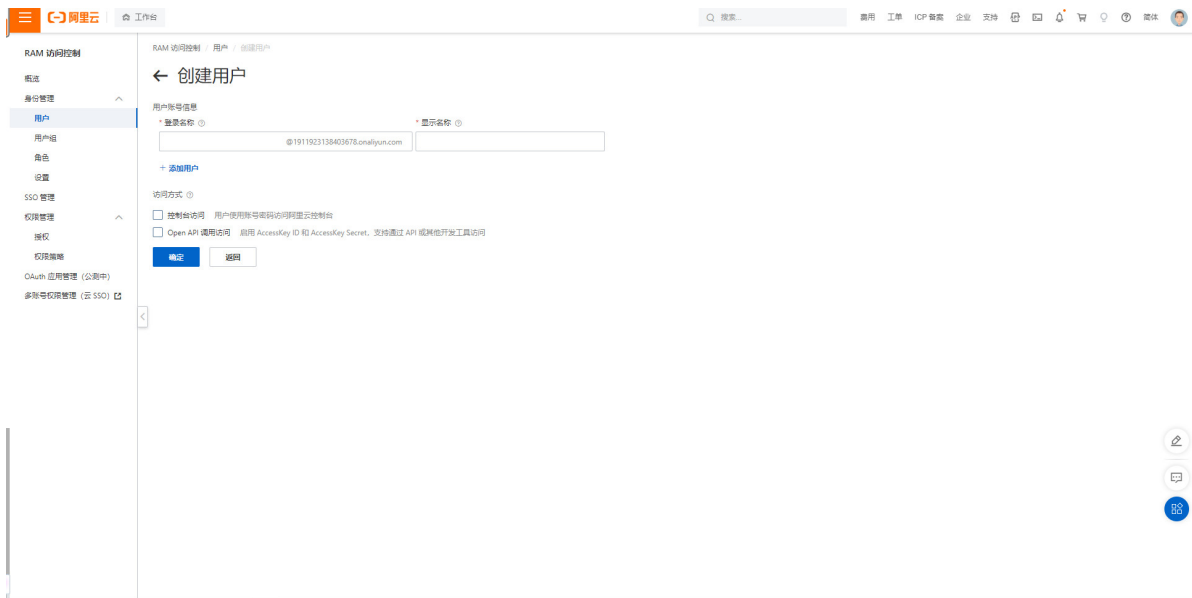
Note: 您的验证码为: `${code}`, 请勿泄露于他人! 中一定要使用 `${code}` 占位以便生成的验证码填充此位置。申请通过的模板会生成一个模板CODE 如官方模板 `SMS_245210276`, 这个code后面编程中需要使用。

Step4: AccessKey管理

- 鼠标放置个人头像会出来下拉选项，点击 AccessKey 按钮，出现如下页面，注意选择 开始使用子用户 AccessKey 按钮，这样用户权限不会太大，更加安全。



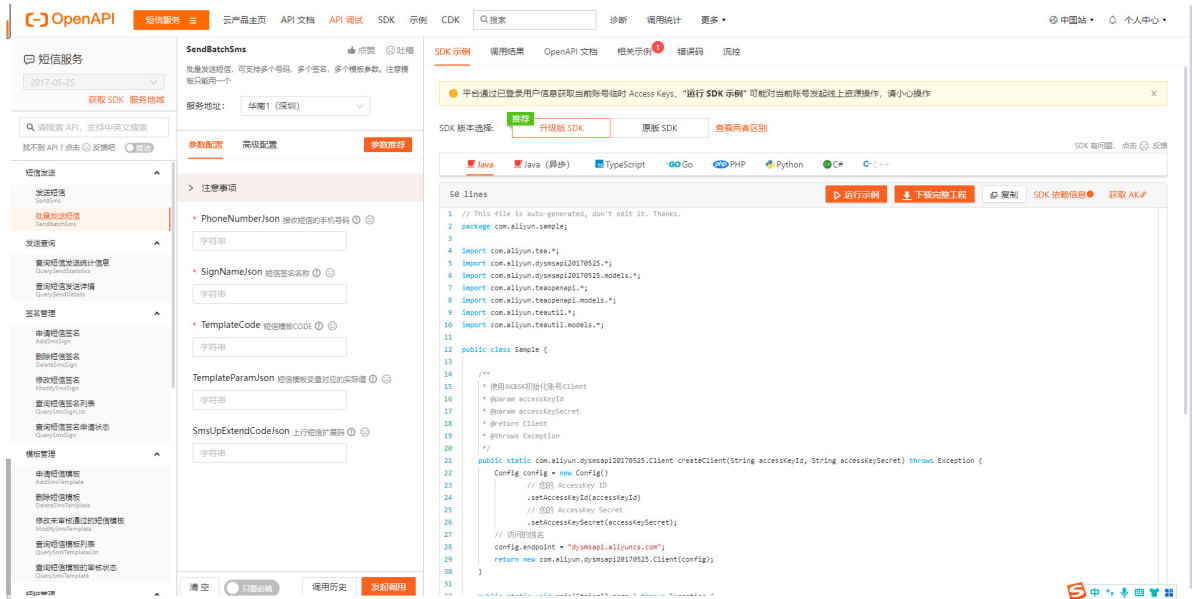
- 点击 **创建用户** 按钮



- 创建用户会生成一个账号和密码，账号是 AccessKey ID 内容，密码是： AccessKey Secret 的内容，记住妥善保管这个账号密码，如果其他人知道你的账号密码，就可以使用你的服务，造成经济损失。

Step5: 调用API编程实现

Note: 可以使用 OpenAPI 进行短信编程学习，并且有代码实例，很好用。



当然也可以是自己参考OpenAPI文档。

个人示例: springboot中使用短息服务

- 首先需导入相关依赖
- 编写 SMSUtils 短息发送工具类
- 随机验证码工具类 validateCodeUtils
- 实现sendMsg方法

```

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.5.16</version>
</dependency>

<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-dysmsapi</artifactId>
  <version>2.2.1</version>
</dependency>

```

```

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.dysmsapi.model.v20170525.SendSmsRequest;
import com.aliyuncs.dysmsapi.model.v20170525.SendSmsResponse;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.profile.DefaultProfile;

/**
 * 短信发送工具类
 */
public class SMSUtils {

    /**
     * 发送短信
     * @param signName 签名
     * @param templateCode 模板
     * @param phoneNumbers 手机号
     * @param param 参数
     */
    public static void sendMessage(String signName, String templateCode, String
phoneNumbers, String param){
        DefaultProfile profile = DefaultProfile.getProfile("cn-hangzhou",
"accesskeyID", "accesskey sercrt");
        IAcsClient client = new DefaultAcsClient(profile);

        SendSmsRequest request = new SendSmsRequest();
        request.setSysRegionId("cn-hangzhou");
        request.setPhoneNumbers(phoneNumbers);
        request.setSignName(signName);
        request.setTemplateCode(templateCode);
        request.setTemplateParam("{\"code\":\""+param+"\"}");
        try {
            SendSmsResponse response = client.getAcsResponse(request);
            System.out.println("短信发送成功");
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}

```

```

import java.util.Random;

/**

```

```

* 随机生成验证码工具类
*/
public class ValidateCodeUtils {
    /**
     * 随机生成验证码
     * @param length 长度为4位或者6位
     * @return
     */
    public static Integer generateValidateCode(int length){
        Integer code = null;
        if(length == 4){
            code = new Random().nextInt(9999);//生成随机数，最大为9999
            if(code < 1000){
                code = code + 1000;//保证随机数为4位数字
            }
        }else if(length == 6){
            code = new Random().nextInt(999999);//生成随机数，最大为999999
            if(code < 100000){
                code = code + 100000;//保证随机数为6位数字
            }
        }else{
            throw new RuntimeException("只能生成4位或6位数字验证码");
        }
        return code;
    }

    /**
     * 随机生成指定长度字符串验证码
     * @param length 长度
     * @return
     */
    public static String generateValidateCode4String(int length){
        Random rdm = new Random();
        String hash1 = Integer.toHexString(rdm.nextInt());
        String capstr = hash1.substring(0, length);
        return capstr;
    }
}

```

```

@PostMapping("/sendMsg")
public R<String> sendMsg(@RequestBody User user, HttpSession session){
    // 获取手机号
    String phone = user.getPhone();
    if (StringUtils.isEmpty(phone)){
        // 生成随机的四位验证码
        String code = ValidateCodeUtils.generateValidateCode(4).toString();
        log.info(code);
        // 调用阿里云提供的短信服务API完成发送短信
        SMSUtils.sendMessage("外卖系统", "", phone, code);
        // 将生成的验证码保存起到Session
        session.setAttribute(phone, code);

        return R.success("手机验证码发送成功！");
    }

    return R.error("短信发送失败！");
}

```

