



链滴

# Spring Boot 注册登录 demo 使用 Spring Security 和发送短信验证码

作者: [king1234](#)

原文链接: <https://ld246.com/article/1648022937700>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

@[TOC]

# 1、前言

接上文 [SpringBoot整合Spring Security实现自定义页面以及验证码登录](#)

## 环境版本说明

MongoDB: 5.0.6

SpringBoot: 2.5.3

JDK: 1.8

## 2、腾讯云短信api

### 介绍

本demo使用免费赠送的腾讯云短信

<https://curl.qcloud.com/a18eCzrO> 短信试用有100个

<https://console.cloud.tencent.com/smsv2>在这里通过模板还会赠送100个

如果不够可以购买

如果觉得短信验证麻烦可换成邮箱验证码

## 3、主要代码

完整代码链接放文末

### pom.xml

这里建议根据api文档更换jar版本

<https://console.cloud.tencent.com/api/explorer?Product=sms>

```
<!-- 邮件依赖 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
<!-- https://mvnrepository.com/artifact/com.tencentcloudapi/tencentcloud-sdk-java-sms -->
<dependency>
  <groupId>com.tencentcloudapi</groupId>
  <artifactId>tencentcloud-sdk-java-sms</artifactId>
  <version>3.1.468</version>
</dependency>
```

### application.yml

sendSms:

SecretId: 你的账号API 密钥id

SecretKey: 你的账号API 密钥

SmsSdkAppId: 你的短信应用id

TemplateId: 你的短信模板id

## 获取SecretId 和 SecretKey

这里的账号api密钥 <https://console.cloud.tencent.com/cam>

快速创建账户，或者直接使用主账号

如图 访问方式使用：编程访问；用户权限：使用预设的短信全读全写，就够了

新建自定义策略

短信

策略列表 (共2条, 已选择1条)

策略名	描述	引用次数	策略类型
<input checked="" type="checkbox"/> QcloudSMSFullAccess	短信服务 (SMS) 全读写访...	1	预设策略
<input type="checkbox"/> QcloudSMSReadOnlyAccess	短信服务 (SMS) 只读访问...	0	预设策略

支持按住 shift 键进行多选

自助管理 API 密钥  自助管理 MFA 设备

确定 取消

CSDN @Hi-King

快速新建用户

什么是快速创建子用户?  
您将通过当前流程快速创建一个或多个子用户，该子用户默认拥有随机密码可登录控制台，拥有AdministratorAccess全局权限，在验证消息渠道后将默认接收腾讯云发送给你的全部消息。若您需要对上述默认内容进行调整，可点击 进行编辑。

- 子用户登录使用用户名，不支持中文，用户名一经确定将无法更改
- 登录密码用于子用户登录控制台，子用户获取到登录密码后可通过 [子用户登录链接](#) 进行登录
- 为保障子账号的账户安全和信息有效接收，子账号在登录时将被要求绑定和验证手机

用户名	访问方式	用户权限	操作
test	编程访问	QcloudSMSFullAccess	删除

新增用户 (单次最多创建10个用户)

需要重置密码  用户必须在下次登录时重置密码

创建用户

CSDN @Hi-King

## 成功新建用户

您已经成功新建用户，用户基础信息如下所示。您可以点击复制或发送该消息至指定邮箱来传达给用户。您也可以[下载](#)当前新建用户的所有信息。

### 用户信息

用户名	密钥	操作
test	SecretId: xx SecretKey: xx	<a href="#">复制</a> <a href="#">发送至</a>

### 权限信息

策略名	策略描述
QcloudSMSFullAccess	短信服务 (SMS) 全读写访问权限

共 1 条

10 / 页

CSDN @Hi-King

这样就得到了 SecretId 和 SecretKey

## 获取SmsSdkAppId 和 TemplateId

去短信--> 应用管理-->应用列表获取你的应用id SmsSdkAppId 没有就新建一个  
<https://console.cloud.tencent.com/smsv2/app-manage>

短信

概览

快速入门

国内短信

国际/港澳台短信

统计分析

套餐包管理

营销管理

应用管理

应用列表

基础配置

通知与告警

安全设置

通用管理

应用列表

如果您希望与工程师即时沟通交流，欢迎通过腾讯云短信小助手联系我



- 1. 创建应用可用于个性化管理短信发送任务，例如设置不同的发送频率和发
- 2. 点击下方已创建应用查看应用详情，调用短信API发送短信时需要使用SD

创建应用

验证码

SDKAppID: [redacted]

标签: 无

停用

共 2 条

CSDN @Hi-King

去短信--> 国内短信-->正文模板管理 获取你审核通过的模板id Templateld 没有就新建一个等待审通过

腾讯云 总览 云产品

短信

- 概览
- 快速入门
- 国内短信
  - 签名管理
  - 正文模板管理**
  - 群发短信
- 国际/港澳台短信
- 统计分析
- 套餐包管理
- 营销管理
- 应用管理
- 通用管理

### 正文模板管理

1. 国内短信由签名+正文组成，签名符号为【】（注：全角），发送短信内容时必须带签名。  
 2. 短信签名和模板提交后，预计2小时完成审核。审核时间：周一至周日9:00-23:00(法定  
 3. 腾讯云短信服务严禁发送金融营销、赌博、诈骗、淫秽、色情、暴力等违规信息，如您  
 集犯罪线索（包括但不限于：腾讯云实名认证信息等）并移交公安机关处理，详情请查  
 4. 受工信部及运营商管控政策影响，三大运营商近期严格管控营销类短信发送，在此期间

[创建正文模板](#) [了解正文模板审核标准，如有疑问，可咨询腾讯云短信小助手](#)

关键字  模板类型

ID	模板类型	模板名称	内容
[REDACTED]	普通短信	注册验证码	您正在申请手机注册效!
1 [REDACTED]	普通短信	发送验证码 固定过期时间	您正在申请手机注册效! 请勿泄露给他息。
[REDACTED]	普通短信	注册	{1}您正在申请手机内有效! 请勿泄露信息。

共 3 条

CSDN @Hi-King

## TxSmsServiceImpl.java 发送短信

```
package com.king.security.service;
```

```
import com.king.security.util.MyException;
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.sms.v20210111.SmsClient;
import com.tencentcloudapi.sms.v20210111.models.SendSmsRequest;
import com.tencentcloudapi.sms.v20210111.models.SendSmsResponse;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Service;
```

```

/**
 * @program: springboot
 * @description: 发送短信
 * @author: King
 * @create: 2022-03-17 15:12
 */
@Service
public class TxSmsServiceImpl {
    @Value(value = "${sendSms.SecretId}")
    private String secretId; //
    @Value(value = "${sendSms.SecretKey}")
    private String secretKey;
    @Value(value = "${sendSms.SmsSdkAppId}")
    private String SmsSdkAppId;
    @Value(value = "${sendSms.TemplateId}")
    private String TemplateId;
    private final Logger logger = LoggerFactory.getLogger(this.getClass());
    //更改短信模板id
    public void setTemplateId(String templateId) {
        TemplateId = templateId;
    }

    /**
     * 发送短信
     * @param phoneNumber 电话号码
     * @param smsCode 验证码
     * @return
     */
    public String sendSmsCode(String phoneNumber, String smsCode) throws MyException {
        try {
            Credential cred = new Credential(secretId, secretKey);
            HttpProfile httpProfile = new HttpProfile();
            httpProfile.setEndpoint("sms.tencentcloudapi.com");
            // 实例化一个client选项, 可选的, 没有特殊需求可以跳过
            ClientProfile clientProfile = new ClientProfile();
            clientProfile.setHttpProfile(httpProfile);
            // 实例化要请求产品的client对象,clientProfile是可选的
            SmsClient client = new SmsClient(cred, "ap-guangzhou", clientProfile);
            // 实例化一个请求对象,每个接口都会对应一个request对象
            SendSmsRequest req = new SendSmsRequest();

            req.setPhoneNumberSet(new String[]{phoneNumber});
            req.setSmsSdkAppId(SmsSdkAppId);
            req.setTemplateId(TemplateId);
            //SignName – 短信签名内容, 使用 UTF-8 编码, 必须填写已审核通过的签名, 例如: 腾
云,
            // 签名信息可前往 [国内短信](https://console.cloud.tencent.com/smsv2/csms-sign)
            // 的签名管理查看。 发送国内短信该参数必填。
            req.setSignName("花草信息个人网");
            //对应短信模板的内容 测试只支持数字字符串
            //您正在申请手机注册, 验证码为: {1}, {2}分钟内有效!
            String[] templateParamSet1 = {smsCode, "30"};
            req.setTemplateParamSet(templateParamSet1);

```

```

        // 返回的resp是一个SendSmsResponse的实例, 与请求对象对应
        SendSmsResponse resp = client.SendSms(req);
        // 输出json格式的字符串回包
        logger.info(SendSmsResponse.toJsonString(resp));

        return resp.getSendStatusSet()[0].getMessage();
    } catch (TencentCloudSDKException e) {
        logger.error(e.toString());
        throw new MyException(e.getMessage());
    }
}
}
}

```

## CodeController.java 发送验证码

```

package com.king.security.controller;

import com.king.security.service.TxSmsServiceImpl;
import com.king.security.util.MyException;
import com.king.security.util.StringUtils;
import com.king.security.vo.ResultObj;
import io.swagger.annotations.Api;
import io.swagger.annotations.ApiImplicitParam;
import io.swagger.annotations.ApiOperation;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import javax.annotation.security.PermitAll;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.util.Date;

/**
 * @program: springboot
 * @description:
 * @author: King
 * @create: 2022-03-18 13:27
 */
@RestController
@PermitAll
@RequestMapping("/api/code")
@Api(value = "验证码接口", tags = "验证码接口")
public class CodeController {
    @Autowired
    private TxSmsServiceImpl txSmsService;

    @ApiOperation(value = "发送注册验证码", httpMethod = "POST")
    @ApiImplicitParam(name = "phone", value = "电话号码", dataType = "string", paramType = "query", example = "13788888888", required = true)
    @PostMapping("/sendSms.do")

```



```
public ResultObj sendCodeRegister(String phone, HttpServletRequest request) {
    try {
        StringUtils.isPhone(phone);
        HttpSession session = request.getSession();
        Date lastDate = (Date) session.getAttribute("code_date");
        if (lastDate != null && (new Date().getTime() - lastDate.getTime()) < (1000 * 120)) {
            return ResultObj.error("别点了, 等" + (120 - ((new Date().getTime() - lastDate.getTi
e()) / 1000)) + "秒后再点");
        } else {
            String code = StringUtils.getInt(6);
            session.setAttribute("code", code);//存验证码到session
            session.setAttribute("code_date", new Date());//存当前时间到session
            return ResultObj.ok(txSmsService.sendSmsCode(phone, code));
        }
    } catch (MyException e) {
        return ResultObj.error(e);
    }
}
}
```

## 4、效果展示

# xxx 注册

XXXXXXXXXXXXXXXXXX

Hello\_World1aa

信息

×



成功!

确定

同意用户隐私协议



注册

社交账号注册



用已有帐号登入

CSDN @Hi-King

```
2022-03-19 01:26:11.100 INFO 15644 --- [nio-9090-exec-1] c.k.security.service.TxSmsServiceImpl : {"SendStatusSet":[{"SerialNo":"00000000000000000000000000000000",
"PhoneNumber":"+8613900000000", "Fee":1, "SessionContext":"","Code":"00", "Message":"send success", "IsoCode":"CN"}, {"RequestId":"0-501001-0711-4006-aa0f-000000000000"}]}
Hibernate: select user0_.id as id1_0_, user0_.create_time as create_t2_0_, user0_.email as email3_0_, user0_.flag as flag4_0_, user0_.head_img as head_img5_0_, user0_.name as name6_0_,
user0_.password as password7_0_, user0_.phone as phone8_0_, user0_.role as role9_0_ from user user0_ where user0_.name=?
Hibernate: insert into user (create_time, email, flag, head_img, name, password, phone, role) values (?, ?, ?, ?, ?, ?, ?, ?)
```

id	create_time	email	flag	head_img	name	password	phone	role
1	(Null)	(Null)	(Null)	(Null)	admin	\$2a\$10\$GLI	(Null)	0
2	(Null)	(Null)	(Null)	(Null)	user	\$2a\$10\$nz3	(Null)	1
3	(Null)	(Null)	(Null)	(Null)	king	\$2a\$10\$qoc	(Null)	2
4	(Null)	(Null)	(Null)	(Null)	Hello_World	\$2a\$10\$88c		1
5	(Null)	(Null)	(Null)	(Null)	Hello_World1aa	\$2a\$10\$yGc		1

## 5、源码地址

Github <https://github.com/KingJin-web/springboot>

Gitee <https://gitee.com/KingJin-web/springboot>