

# 验证码服务使用指南

## 1 部署验证码服务

### 1.1 基础环境

Java 1.8+

Maven3.3.9+

### 1.2 安装Redis

参考“Redis安装指南.pdf”

### 1.3 部署验证码服务

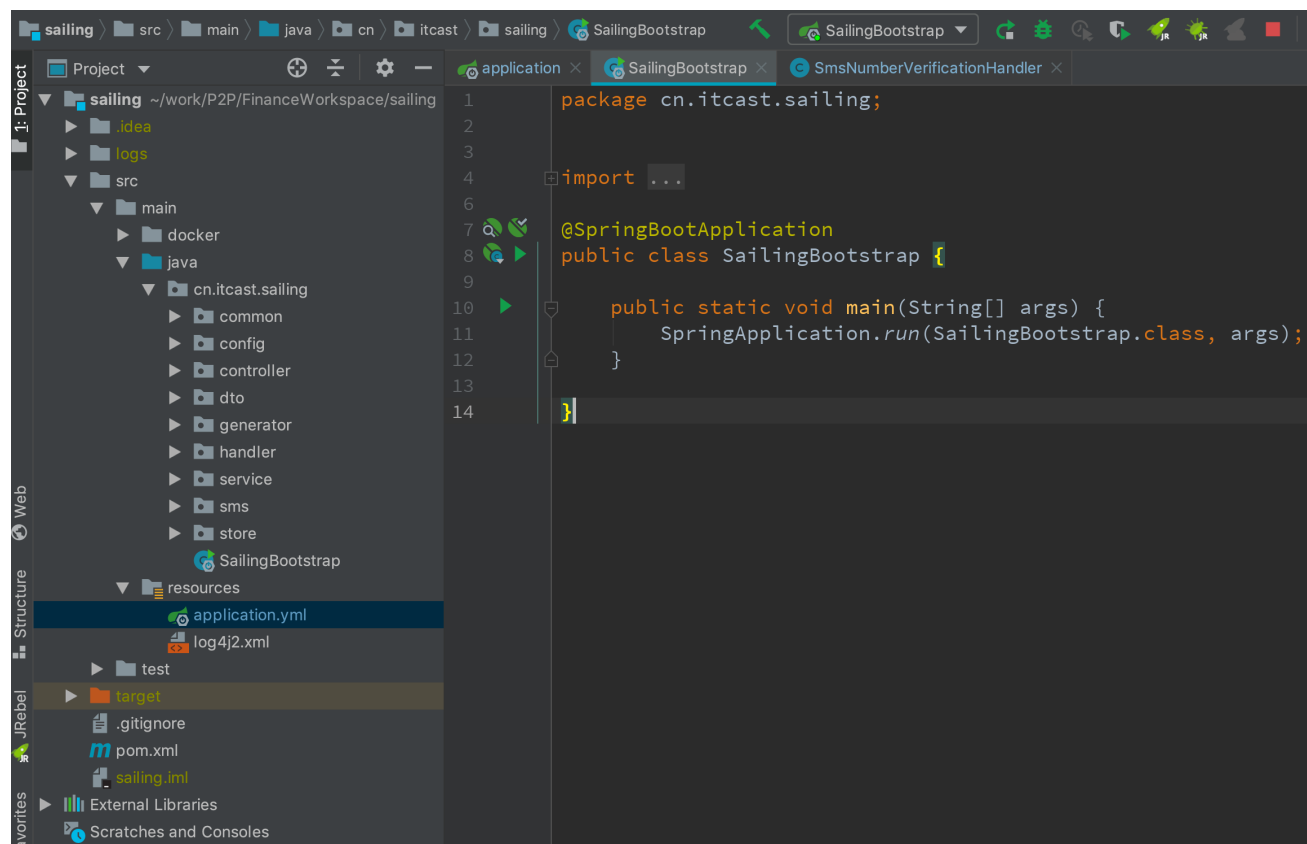
#### 1.3.1 下载源码

使用git从远程下载验证码服务代码(开源)。

```
git clone https://github.com/fightingape/sailing.git
```

#### 1.3.2 使用idea打开项目

使用idea打开上一步下载的sailing目录，下图是sailing在idea的工程结构。



### 1.3.3 开通腾讯云短信服务

本服务中的短信发送使用了腾讯云短信服务，生产环境需要注册腾讯云开通短信服务：

#### 1) 开通腾讯云短信服务

<https://cloud.tencent.com/product/isms/getting-started>

1. 开通短信服务成功后，获取 SDKAppID 和 AppKey  
云短信应用 **SDKAppID** 和 **AppKey** 可在 [短信控制台](#) 的应用信息里获取。如您尚未添加应用，请登录 [短信控制台](#) 添加应用。
2. 申请签名并确认审核通过  
一个完整的短信由短信**签名**和**短信正文内容**两部分组成，短信**签名**需申请和审核，**签名**可在 [短信控制台](#)的相应服务模块【内容配置】中进行申请，详细申请操作请参见 [创建签名](#)。发送国际短信时，允许不携带签名。
3. 申请模板并确认审核通过  
短信或语音正文内容**模板**需申请和审核，**模板**可在 [短信控制台](#) 的相应服务模块【内容配置】中进行申请，详细申请操作请参见 [创建正文模板](#)。

#### 2) 配置短信参数

1. 打开项目中sailing/src/main/resources/application.yml
2. 将腾讯云短信服务获取到参数做如下配置：

```
sms:
  qcloud:
    appId: 14000000
    appKey: 36ff7sd14d0d2342344cdf335cc25052
    templateId: 234323
    sign: 签名
```

appId：上边第1步开通短信服务成功后获取SDKAppID

appKey：上边第1步开通短信服务成功后获取AppKey

sign：上边第2步申请签名

templateId：上边第3步申请模板

### 1.3.5 控制台获取验证码

如果暂时没有申请到腾讯云短信服务，可以临时将验证码输出到控制台。

修改sailing/src/main/java/cn/itcast/sailing/handler/SmsNumberVerificationHandler.java

```

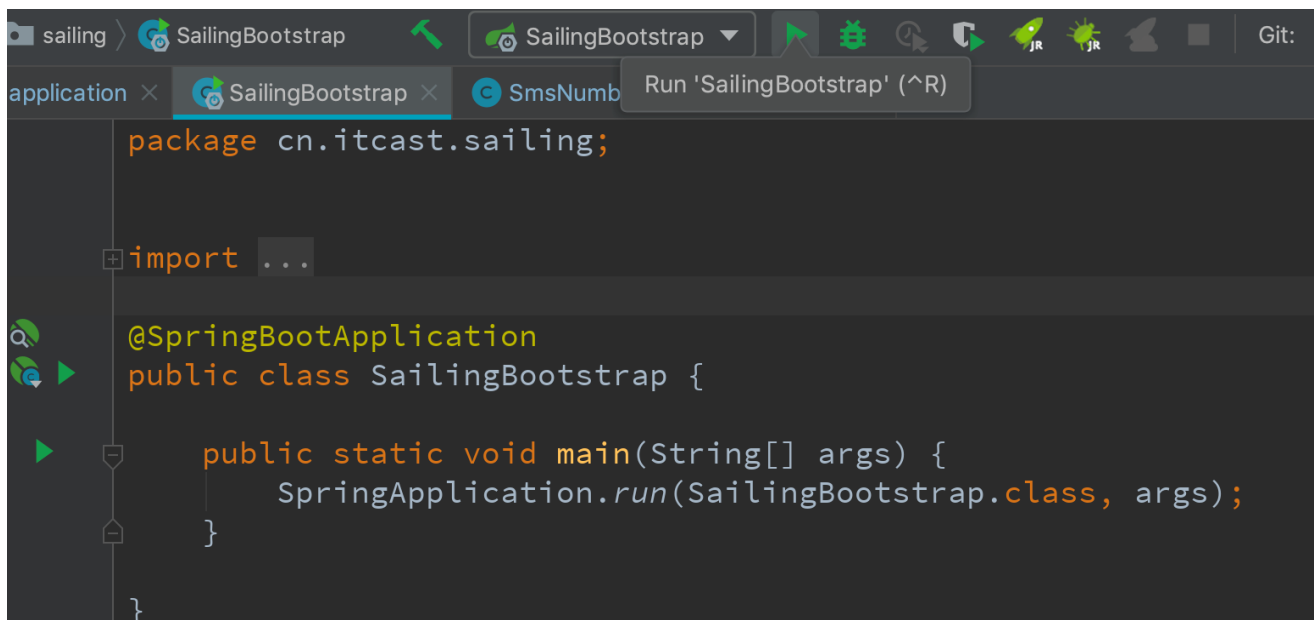
@Override
String confusion(Map<String, Object> payload, String key, String code) {
    String mobile = String.valueOf(payload.get("mobile"));

    // 使用腾讯云发送短信
    //smsService.send(mobile, code, getEffectiveTime());

    // 测试使用, 在控制台输出验证码
    smsService.sendOnConsole(mobile, code, getEffectiveTime());
    return null;
}

```

### 1.3.6 启动服务



## 2 使用验证码服务

### 2.1 发送验证码

1. 接口说明：<http://localhost:56085/sailing/swagger-ui.html>

参数列表如下：

参数名	参数类型	描述
name	string(query)	业务名称(使用何种方式的验证码, 可选的有sms, img, mail等)
payload	application/json	业务携带参数, 如手机号、邮箱
effectiveTime	string(query)	验证信息有效期(秒)

POST /generate 生成验证信息

生成验证信息

Parameters Try it out

Name	Description
effectiveTime string (query)	验证信息有效期(秒)
name * required string (query)	业务名称
payload * required (body)	业务携带参数, 如手机号, 邮箱

Example Value | Model

string

Parameter content type  
application/json

2. 点击'Try it out', 填写参数进行测试

POST /generate 生成验证信息

生成验证信息

Parameters Cancel

Name	Description
effectiveTime string (query)	验证信息有效期(秒)
name * required string (query)	业务名称
payload * required (body)	业务携带参数, 如手机号, 邮箱

Example Value | Model

{"mobile": 13612341234}

Edit

Parameter content type  
application/json

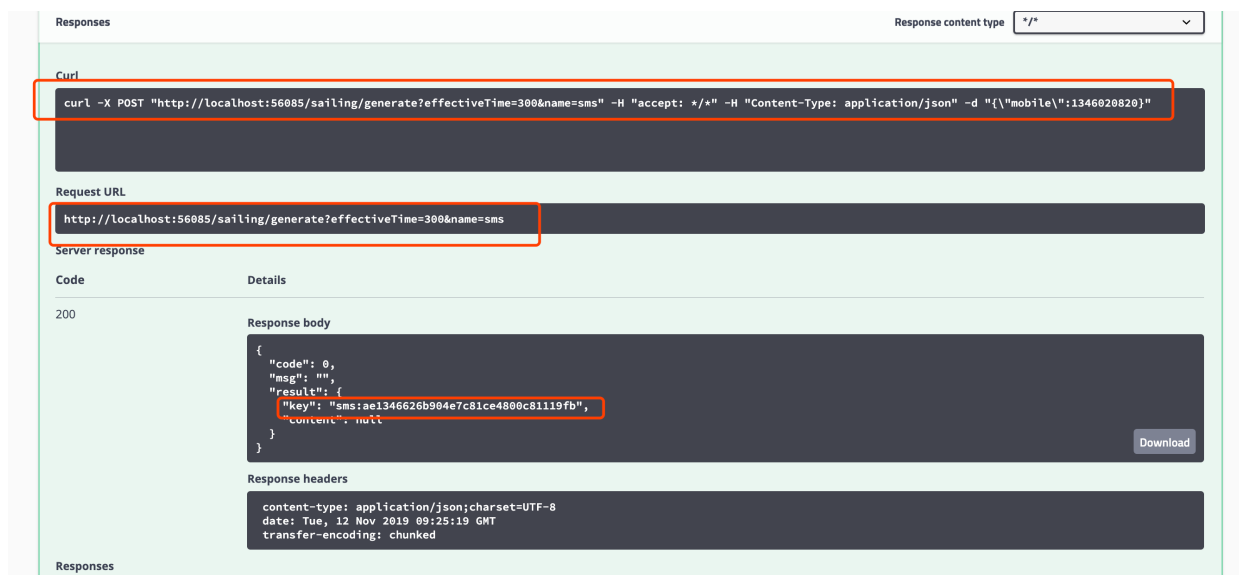
Execute Clear

参数说明：

name:业务名称，短信的业务名称为"sms"。

payload：发送短信时这里要填写mobile，json格式。

3. 查看返回的此次验证key：校验验证码时需要携带验证key和用户输入的验证码



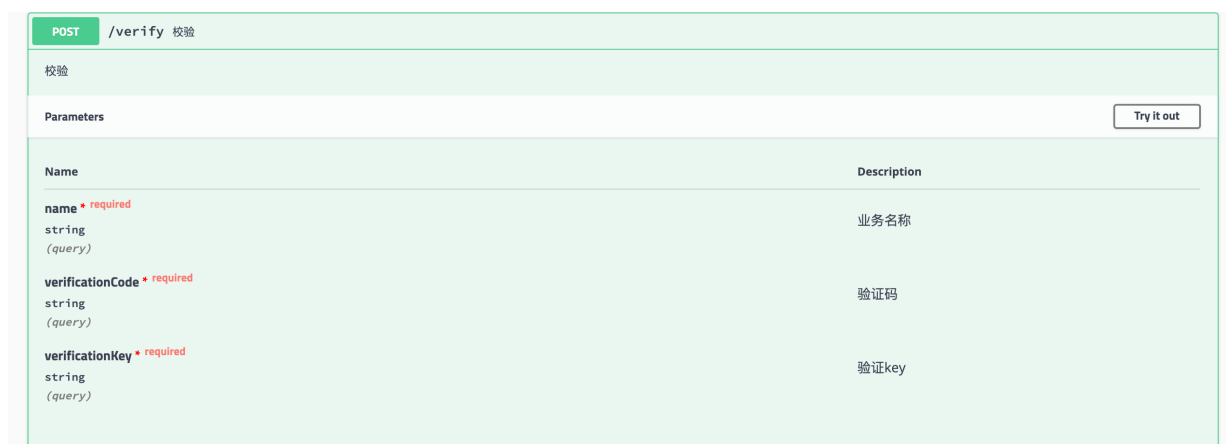
发送验证码成功，可通过手机查看，如果在开发阶段可在控制台查看验证码：

INFO [http-nio-56085-exec-1][QCloudSmsService.java:37] - 给手机号xxxxxxxxxxx发送验证码：xxxx

## 2.2 校验验证码

1. 接口说明：<http://localhost:56085/sailing/swagger-ui.html#/verification-controller/verifyUsingPOST>

参数名	参数类型	描述
name	string(query)	业务名称(使用何种方式的验证码，可选有sms, img, mail等)
verificationCode	string(query)	验证码
verificationKey	string(query)	验证key



2. 点击'Try it out', 填写验证码和验证key进行测试

POST

/verify 校验

校验

Parameters

Cancel

Name	Description
<b>name</b> <small>required</small> string (query)	业务名称 <div>sms</div>
<b>verificationCode</b> <small>required</small> string (query)	验证码 <div>039932</div>
<b>verificationKey</b> <small>required</small> string (query)	验证key <div>sms:a963ba330df940ad8896c7e01fa5f2ae</div>

Execute

Clear

name:业务名称，短信的业务名称为"sms"。

verificationCode：验证码，手机接收的验证码，如果在开发阶段通过控制台查看验证码，可在控制台中查看验证码。

verificationKey：发送验证码接口响应的内容，如下：

```
{
  "code": 0,
  "msg": "正常",
  "result": {
    "key": "sms:54bdd3b721f3419aa9445e85d45b329f",
    "content": null
  }
}
```

### 3. 查看校验结果：返回true则为校验通过

Curl

curl -X POST "http://localhost:56085/sailing/verify?name=sms&verificationCode=039932&verificationKey=sms%3Aa963ba330df940ad8896c7e01fa5f2ae" -H "accept: \*/\*"

Request URL

http://localhost:56085/sailing/verify?name=sms&verificationCode=039932&verificationKey=sms%3Aa963ba330df940ad8896c7e01fa5f2ae

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{   "code": 0,   "msg": "",   "result": true }</pre></div><div>Download</div></div> <div><div>Response headers</div><div><pre>content-type: application/json;charset=UTF-8 date: Tue, 12 Nov 2019 09:56:12 GMT transfer-encoding: chunked</pre></div></div>

Responses