# 短信验证（Android）

# 介绍

**什么是短信验证：**

短信验证是企业给消费者（用户）的一个凭证，通过短信内容的码来验证身份。

**短信验证的用处：**

在现如今人手好几只手机，至少一张sim卡的情况下，使用短信验证来确定用户的身份变得高效和准确，较少了很多的非法访问的情况。

手机app注册或者登录用户的时候，手机号作为登录的账户也变得越来越广泛了。

**短信平台的诞生：**

如何实现短信登录呢，自己写服务器，沟通短信服务商，会非常忙的。因此，就出现了很多的辅助平台，例如bmob 、mob等。可以使得引用开发得到便利，当然这些平台总是需要利益驱使的。

**短信验证的流程：**

如下是mob的一张图：

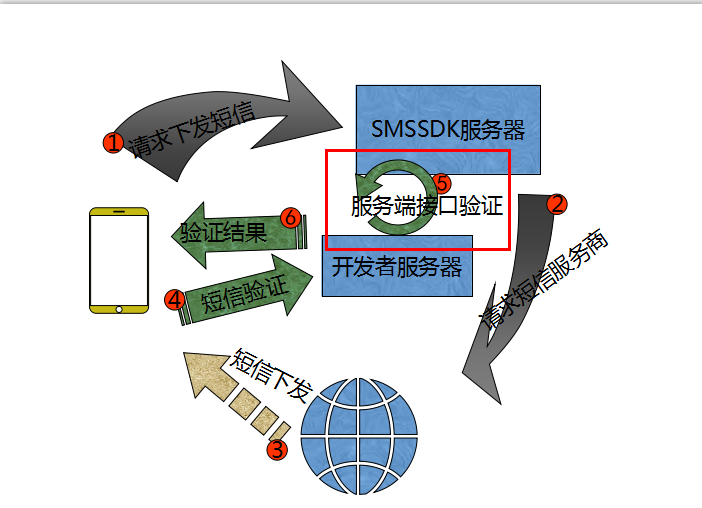


图1-1 短信验证图

主要可以分下面四步：

1. 手机请求短信验证服务器
2. 短信验证服务器请求短信服务商发出短信
3. 手机接收到短信，将验证码请求验证服务器
4. 验证服务器返回验证结果

# 集成环境

接下来使用mob作为短信验证平台来实现短信验证。

**注册mob开发者账户**

官网（http://sms.mob.com/）注册一波就好了，简单操作。

**创建一个应用**

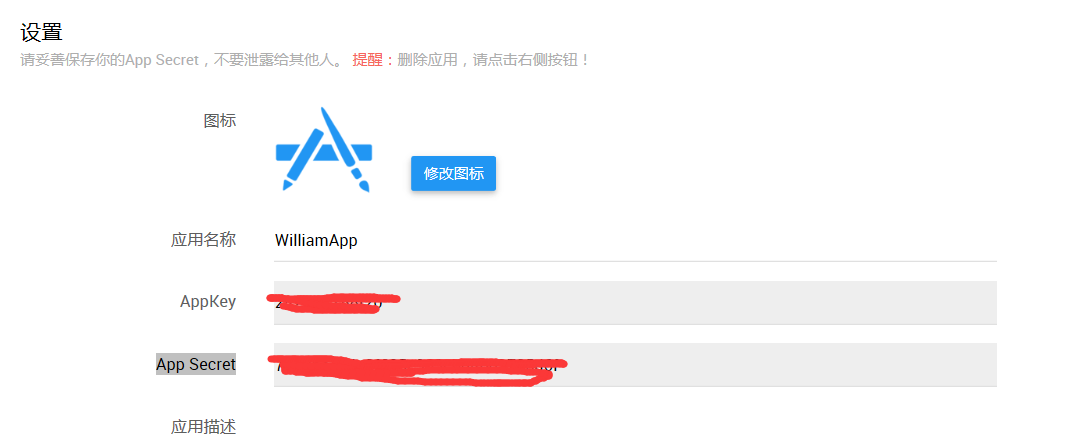
****

**图2-1 创建应用**

写一个应用的名称就可以，暂时不需要其他信息。

在设置里面查看创建应用的AppKey和App Secret ，访问服务器需要用到的两个验证应用的码。

主要就是要这两个码。

****

**图2-2 设置**

**接下来就是集成项目了：**

**项目集成（android为例）**

1）sdk导入

在根build中，buildscript下的repositories添加：

maven{url "http://mvn.mob.com/android" }

buildscript下的dependencies添加：

classpath 'com.mob.sdk:MobSDK:+'

总的代码：

buildscript {  
 repositories {  
 maven{url "http://mvn.mob.com/android" }  
 }  
 dependencies {

classpath 'com.mob.sdk:MobSDK:+'  
 }  
}

2）插件的导入

如果项目中没有Application ，mob会自动将com.mob.MobApplication作为Application类。插件可以自动绑定，appkey和appsecret。

在短信代码编写的gradle中加入如下代码：

apply plugin: 'com.mob.sdk'  
// 在MobSDK的扩展中注册SMSSDK的相关信息  
MobSDK {  
 appKey "appkey码"  
 appSecret "appSecret码"  
 SMSSDK {}  
}

上面的两个主要的识别码需要对应创建的引用平台给的编码。

如果存在application，需要在oncreate中添加：

MobSDK.*init*(this);

记得在AndroidManifest.xml中的application添加;

tools:replace="android:name"

不然会编译不通过。

好了这样就集成好了。

如果需要手动集成的话访问下网址：

<http://wiki.mob.com/smssdk-old-android/>

# 使用

**1）短信验证申请**

**点击验证按钮，请求发送短信至手机。**

SMSSDK.*registerEventHandler*(new EventHandler() {  
 public void afterEvent(int event, int result, Object data) {  
 if (result == SMSSDK.*RESULT\_COMPLETE*) {  
 // *TODO 处理验证成功的结果*  if (event == SMSSDK.*EVENT\_SUBMIT\_VERIFICATION\_CODE*) {  
 //提交验证码成功  
 Log.*i*("EventHandler", "提交验证码成功");  
 } else if  
 (event == SMSSDK.*EVENT\_GET\_VERIFICATION\_CODE*) { //获取验证码成功  
 Log.*i*("EventHandler", "获取验证码成功");  
 } else if  
 (event == SMSSDK.*EVENT\_GET\_SUPPORTED\_COUNTRIES*) {  
 //返回支持发送验证码的国家列表  
 Log.*i*("EventHandler", "返回支持发送验证码的国家列表");  
 }  
 } else {  
 // *TODO 处理错误的结果*   
 }  
 }  
});

SMSSDK.*getVerificationCode*(国家编码, 手机号);

首先注册handler，然后发送请求*g*etVerificationCode后会调用handler中afterEvent方法，返回参数i1对应事件结果。

**2）验证短信**

获取到验证码后，填写到合适位置然后访问：

SMSSDK.*submitVerificationCode*(国家编码, 手机号, 验证码);

实现效果：



代码送上https://github.com/xiaogoudandan/WilliamApp

关注一下公众号，一起进步！

