# 用户中心

三种常见用户登录：传统登陆、短信登陆、第三方登陆。

用户名与密码不是必须的。通过短信也可以快速完成登陆。知道用户在第三方平台上的用户信息可以完成快速登陆。

## 传统登陆

必须输入用户名和密码。

## 短信校验

两种处理方式：

1. 自己发送校验短信，完成用户校验
2. 借助第三方平台完成用户校验

### 1、使用ShareSDK（http://www.mob.com/）

官方文档链接：

<http://wiki.mob.com/android-%e7%9f%ad%e4%bf%a1sdk%e9%9b%86%e6%88%90%e6%96%87%e6%a1%a3/>

注册用户，并登录，进入后台。



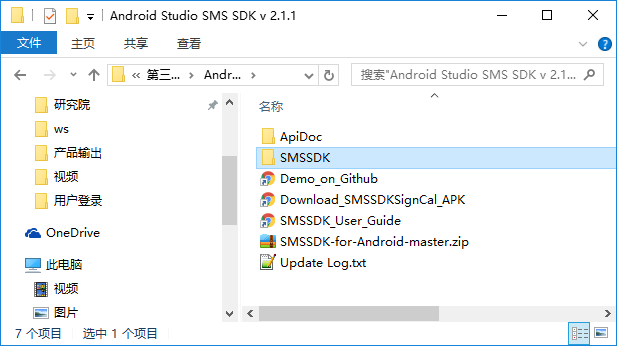
### 2、开发者平台添注册并添加应用

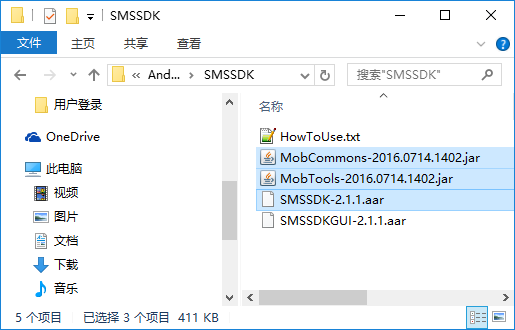


App Key：16db05f4edda0

App Secret：f5f8e5c52e11337353979972c1b08cae

### 3、下载并解压





实际开发中都会自己定制相关界面，所以我们在使用选中的三个文件即可。

### 4、引入第三方工具

在HowToUse.txt中有关于如何配置的说明。

将以上文件按需放入Android Studio项目所要使用SMSSDK的Module所在的Libs里面，再在Module的build.gradle里面将libs加入仓库（repositories），

然后添加依赖（dependencies ）示例如下：

repositories{

flatDir{

dirs 'libs' //就是你放aar的目录地址

}

}

dependencies {

....//你的其他依赖

compile name:'SMSSDK-<version>',ext:'aar'

compile name:'SMSSDKGUI-<version>',ext:'aar'

}

### 5、配置AndroidManifest.xml

权限：

<uses-permission android:name="android.permission.READ\_CONTACTS" />

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.RECEIVE\_SMS" />

<uses-permission android:name="android.permission.READ\_SMS" />

<uses-permission android:name="android.permission.GET\_TASKS" />

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

添加Activity配置

<activity

android:name="com.mob.tools.MobUIShell"

android:theme="@android:style/Theme.Translucent.NoTitleBar"

android:configChanges="keyboardHidden|orientation|screenSize"

android:windowSoftInputMode="stateHidden|adjustResize"/>

### 6、核心步骤：

1. 权限校验: SMSUtil.*checkPermission*(**this**);
2. 初始化工具: SMSSDK.*initSDK*(**this**, *APPKEY*, *APPSECRET*, **true**);
3. 注册事件监听：SMSSDK.*registerEventHandler*(**eventHandler**);
4. 获取验证码：SMSSDK.*getVerificationCode*(**"86"**, **phone**);监听事件触发。
5. 发送验证码：SMSSDK.*submitVerificationCode*(**"86"**, **phone**, code.trim());监听事件触发。
6. 注销监听。

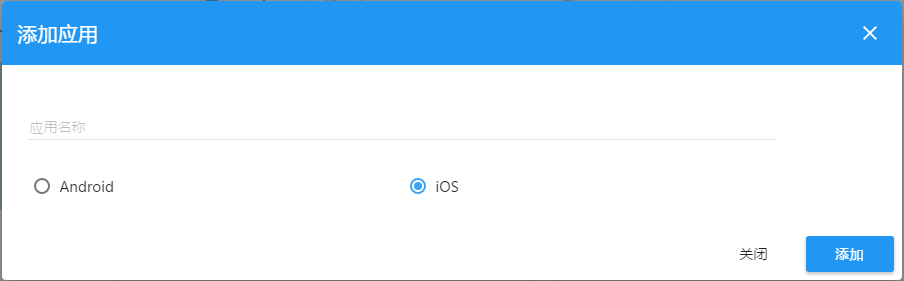
## 第三方登录

### 使用ShareSDK

官方说明文档：

<http://wiki.mob.com/android_%e5%bf%ab%e9%80%9f%e9%9b%86%e6%88%90%e6%8c%87%e5%8d%97/>

注册为其用户，添加自己应用的基本信息（选择系统、填写应用名称）。



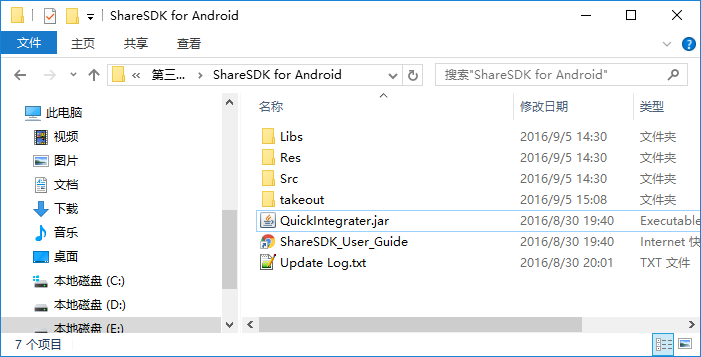
App Key：16d2811eb998a

App Secret：1180a1cef484edf19a7507be1e5ff231

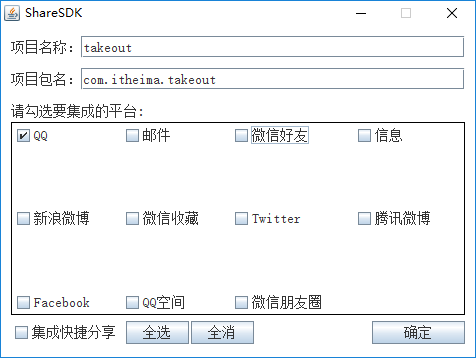
下载SDK



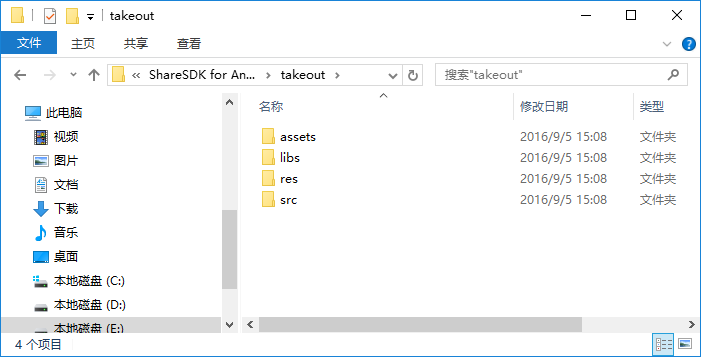
解压



运行：QuickIntegrater.jar



使用QQ完成第三方登录，注意“集成快捷分享”不需要勾选。生产如下内容，拷贝到工程中即可。



修改assets下的ShareSDK.xml文件<ShareSDK AppKey = "16d2811eb998a"/>值为注册应用时结果。

### 2、开发者平台注册并添加应用。

微信开发者平台注册



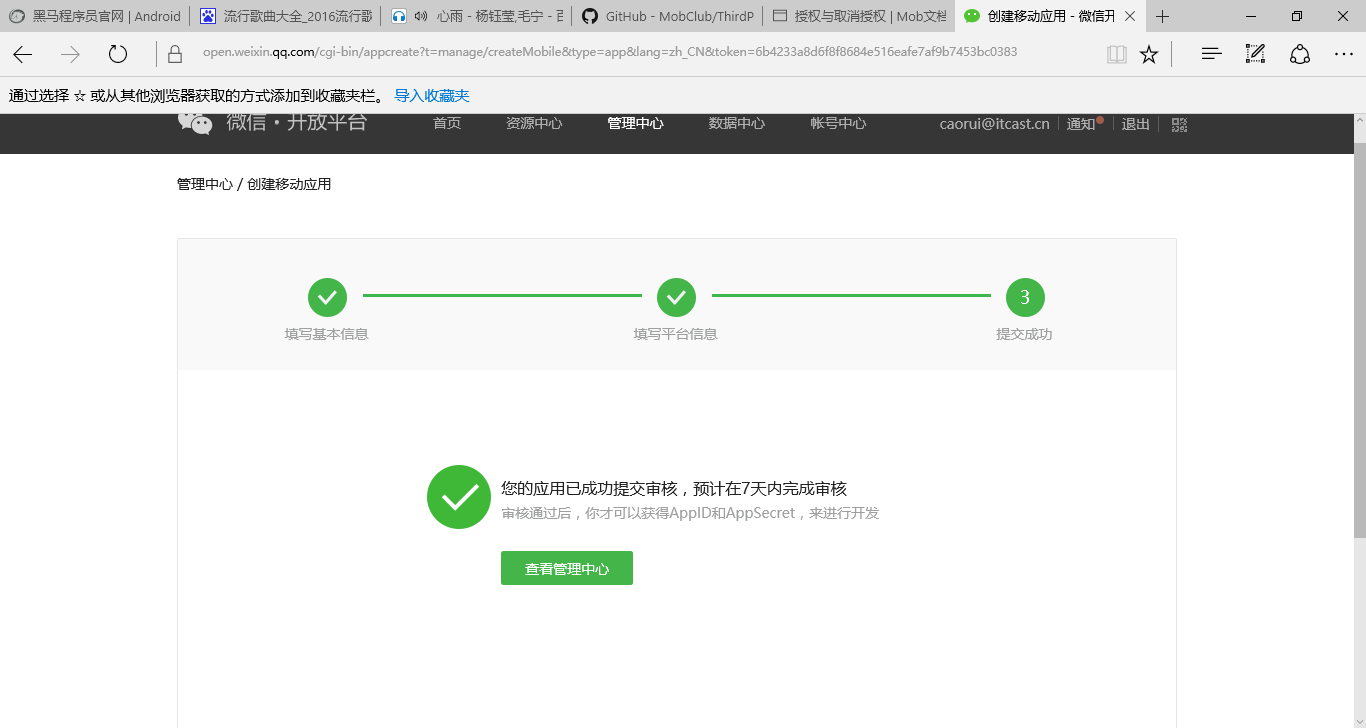




查看应用签名：AS中签名文件是.jks，

使用命令：keytool -list -v -keystore “jks文件路径”

MD5 5A:E4:F2:01:D8:C7:10:59:14:18:0E:1A:EE:7A:70:00



QQ: <http://wiki.connect.qq.com/%E7%A7%BB%E5%8A%A8%E5%BA%94%E7%94%A8%E6%8E%A5%E5%85%A5%E6%B5%81%E7%A8%8B>



### 3、使用ShareSDK官方demo展示处理过程

核心步骤：

// 第一步:初始化

ShareSDK.*initSDK*(**this**);

// 第二步：设置结果回掉监听

plat.setPlatformActionListener(**this**);

// 第三步：SSO：4、SSO (Single Sign-On) SSO授权方式，简单来说就是使用目标平台客户端来完成授权，有客户端的都会优先启用客户端授权，没客户端的则任然使用网页版进行授权。。

plat.SSOSetting(**true**);

// 第四步：要数据，不要功能

plat.showUser(**null**);

// 第五步：回复结果监听处理

onComplete(Platform platform, **int** action,HashMap<String, Object> res)

// 第六步：退出时调用，回收资源，并帮助统计应用反问次数

ShareSDK.*stopSDK*(**this**);

### 4、经验总结

1. 一旦发现应用中会使用到第三方登录的功能必须明确如下内容：
   1. 公司在这些三方平台上都注册开发者了吗？
   2. 应用需要经过不同平台的审核，才能获取到APP ID及相关信息，所以必须提前完成这些工作。否则到项目后期会因外无法快速通过审核而推迟上线进度。
2. 公司的形式注册不同平台开发者，会有一些经济支持，不同平台情况不同。
3. 实际工作中的处理流程:
   1. 我们自己的服务器是否有用户管理系统。
   2. 如果有我们使用第三方登录的目的就是把大平台的用户拿过来放到我们自己的数据库。
   3. 当获取到大平台的数据后，我们需要将数据经过处理：比如用户名我们可以增加一个平台的前缀，后缀使用随机数。比如：平台反馈名称：teacher，处理结果：QQteacher123。将处理后的结果作为我们的临时用户名，如果用户不喜欢可以自行修改。
   4. 第三方登录平台提供的数据可能不能满足我们的需求，所以我们需要在业务进行的过程中不断去完善信息收集。