

# 中国联通沃登录Android平台SDK

版本1.0.3

## 1.名词解释

- 1) NET取号：通过运营商网络自动取号。
- 2) 隐式登录：SDK在后台进行NET取号，取到手机号后以alert形式弹出loading, 并进行授权，授权成功后loading消失，用户不参与交互。若NET取号失败则不进行弹窗。
- 3) 一键登录：SDK在后台进行NET取号，取到手机号后以alert形式弹出授权页面，用户确认登录后授权成功。若NET取号失败则不进行弹授权页面。
- 4) 验证登录：SDK在后台进行NET取号，无论取号成功与否，登录界面都会以modal形式进行推出。
- 5) CLIENT\_ID:第三方用户申请到的AppKey。
- 6) CLIENT\_SECRET:第三方用户申请到的AppSecret。
- 7) REDIRECT\_URI:第三方用户申请时填写的回调地址。

## 2.SDK接入设置



- 1)注册成为开发者
- 2)按要求填写基本信息

管理中心/ 创建应用

1

填写基本信息

2

填写平台信息

3

提交成功

应用名称

请注意，应用审核通过后，应用名称不允许修改。

应用类型

请选择

应用简介

最多80字

应用图片

请上传应用水印图片  
30\*30像素，2M以内，支持PNG、JPG。

图片预览

上传

### 3)填写平台信息

☒ Android 应用

应用下载地址  
(选填)

Android应用商店中的下载页面的地址，不允许直接使用apk包的下载地址，如应用还未上线，可置空，待应用上线后再行修改，地址格式：<https://developer wo.cn>,<http://www.unisk.cn/>,<http://127.0.0.1:8080/>应用名称等

应用签名

用于对当前应用进行二次身份校验，开发者可以使用签名生成工具直接从安装当前应用的手机中获取。应用签名由开发者签名该应用的keystore文件决定。

应用包名

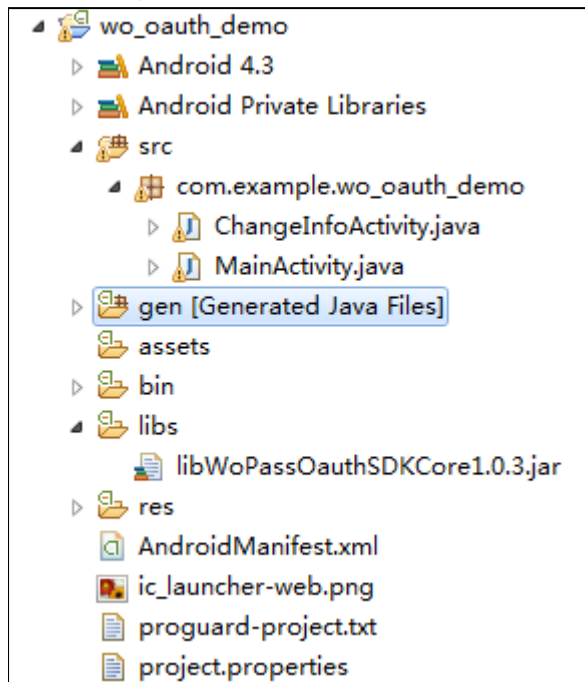
应用在一台设备上的唯一标识，在manifest文件里面声明，该包名应和正式发布应用的包名一致。例如，包名为cn.unisk.open

Scheme

方便APP之间互相调用，通过系统的OpenURI来打开该app，并可以传递一些参数；每个URL必须能唯一标识一个APP。

### 3.SDK接入准备流程:

1) 下载好的libWoPassOauthSDKCore.1.0.3文件添加到libs文件夹内, 如下图:



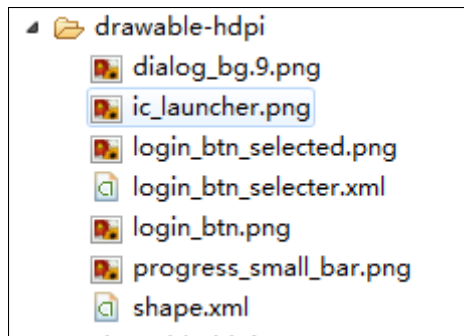
(2) 右键单击工程, 选择Build Path中的Configure Build Path..., 选中Libraries这个tab, 并通过Add Jars...导入工程libs目录下的libWoPassOauthSDKCore1.0.3.jar文件, 注意在demo中用的是测试testLibWoPassOauthSDKCore1.0.3.jar文件, 请不要直接使用。

(3) 在style中添加样式信息引入。

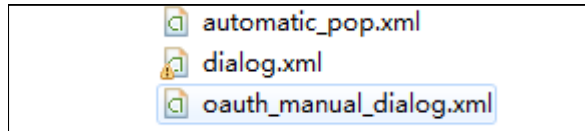
```
<style name="wo_passport_dialog" parent="@android:style/Theme.Dialog">
    <item name="android:windowFrame">@null</item>
    <item name="android:windowIsFloating">true</item>
    <item name="android:windowIsTranslucent">false</item>
    <item name="android:windowNoTitle">true</item>
    <item name="android:background">@android:color/transparent</item>
    <item name="android:windowBackground">@android:color/transparent</item>
    <item name="android:backgroundDimEnabled">true</item>
    <item name="android:windowFullscreen">true</item>
</style>

<style name="Dialog_Fullscreen">
    <item name="android:windowFullscreen">true</item>
    <item name="android:windowNoTitle">true</item>
</style>
```

添加资源文件:



将demo中的这7个资源文件拷贝到工程中相应目录中



将demo中的这个布局文件拷贝到工程中的相应目录中

(4)添加必要的权限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

(5)配置client\_id和client\_secret与重定向url

(6)实现回调接口，用于获取oauth登录的返回的数据

```
implements WoOauthCallbackListener
```

实现这个接口的时候，需要实现三个回调方法：onWoOauthSuccess这个方法，用于请求成功返回数据；onWoOauthFail这个方法，是请求失败的回调方法；onWoOauthMongoliaLayerFinish，这个方法是提供给开发者用于，取消蒙层使用。

## 4.授权模式

SDK中提供了两种授权模式：授权码模式和简化模式。

1)授权码模式：授权码模式集成，第三方用户，必须传入CLIENT\_SECRET参数。

2)简化模式：简化模式集成，第三方用户，不必传入CLIENT\_SECRET参数。

注意：不建议使用授权码模式，暴露应用的CLIENT\_SECRET参数存在安全风险。

## 5.授权方式

SDK提供两种授权方式：验证授权和按级别授权（即旧版本的自动授权）。

## 1)验证授权

这个方式，如果用户手机上安装着沃通行证，会调用SSO方式，进行验证登录。如果，没有安装沃通行证，会进行普通的授权方式。集成方式如下图：

```
/**
 * 验证授权
 * @param mode 用于区分授权码模式与简化模式
 * 授权码模式：需要传入CLIENT_SECRET参数
 * 简化模式：不必传入CLIENT_SECRET参数
 * 注意：不建议使用授权码模式，暴露应用的CLIENT_SECRET存在风险
 */
private void manualAuthorizationMode(String mode) {
    if ("code".equals(mode)) {
        //授权码模式，必须传入CLIENT_SECRET
        WogarOAuth wogarOAuth = new WogarOAuth(context, CLIENT_ID,
            CLIENT_SECRET, REDIRECT_URI);
        wogarOAuth.manualAuthorizationMode(this, true,
            R.style.Dialog_Fullscreen);
    } else if ("token".equals(mode)) {
        //简化模式，不用传入CLIENT_SECRET
        WogarOAuth wogarOAuth = new WogarOAuth(context, CLIENT_ID,
            REDIRECT_URI);
        wogarOAuth.manualAuthorizationMode(this, true,
            R.style.Dialog_Fullscreen);
    }
}
```

## 2.按等级授权

集成这种授权方式，授权形式和第三方应用的安全级别有关。应用的安全级别共分为三个级别：0、1、2三个级别。

应用的安全级别和授权方式对应的关系如下所示：

应用安全级别0：隐式登录授权。

应用安全级别1：一键登录授权。

应用安全级别2：验证登录授权。

集成方式如下图：

```
/**
 * 等级授权
 * @param delaytime 延时调用此方法的时间，单位是秒
 * @param mode 用于区分授权码模式与简化模式
 * 授权码模式：需要传入CLIENT_SECRET参数
 * 简化模式：不必传入CLIENT_SECRET参数
 * 注意：不建议使用授权码模式，暴露应用的CLIENT_SECRET存在风险
 */
private void automaticAuthorizeMode(int delaytime, String mode) {
    if ("code".equals(mode)) {
        //授权码模式，必须传入CLIENT_SECRET
        WogarOAuth wogarOAuth = new WogarOAuth(context, CLIENT_ID,
            CLIENT_SECRET, REDIRECT_URI);
        String state = wogarOAuth.automaticAuthorizeMode(this, screenWidth,
            screenHeight, delaytime, R.style.wo_passport_dialog);
        tv_show.setText(state);
    } else if ("token".equals(mode)) {
        //简化模式，不用传入CLIENT_SECRET
        WogarOAuth wogarOAuth = new WogarOAuth(context, CLIENT_ID,
            REDIRECT_URI);
        String state = wogarOAuth.automaticAuthorizeMode(this, screenWidth,
            screenHeight, delaytime, R.style.wo_passport_dialog);
        tv_show.setText(state);
    }
}
```

自动授权模式，可以设置延迟调用时间，单位是秒。

第三方需要判断用户是否登录，如果未登录，可以启动这个模式，这个会进行net取号，成功后，根据应用的安全级别进行授权登录。

### 3.获取用户信息

可调用下图所示的方法获取用户信息。

```
/**
 * 获取用户信息
 */
protected void getUserInfo() {
    WogarOAuth wogarOAuth = new WogarOAuth(context);
    wogarOAuth.getUserInfo(this, access_token, openid, CLIENT_ID);
}
```

### 6.返回数据格式

授权成功后，均以JSON的形式将数据返回。

1)授权码模式返回数据：

```
{
  "data": {
    "openid": "1535600135507",
    "expires_in": 1812617,
    "refresh_token": "e57e261ba2c75077380e5cb8c9814fa2",
    "access_token": "c6adaf6cfccd654a1a1cb78f3e6fbcbc",
    "code": "0",
    "msg": "成功"
  }
}
```

2)简化模式返回数据：

```
{
  "data": {
    "openid": "1535600135507",
    "access_token": "c6adaf6cfccd654a1a1cb78f3e6fbcbc",
    "expires_in": 1812217,
    "code": "0",
    "msg": "成功"
  }
}
```

3)获取用户信息返回数据：

```
{
  "data": {
    "created_date": "2015-12-22 09:51:47",
    "email": "",
    "nickname": "用户0034",
    "user_id": "1535600135507",
    "avatar": "http://api.life.wobendi.cn/passport/avatar/default_avatar.png",
    "code": "0",
    "msg": "成功"
  }
}
```

用户信息返回数据说明：

avatar:用户头像链接。

created\_data:沃通行证帐号创建时间。

email:电子邮箱。

nickname:用户昵称。

user\_id;用户唯一标示