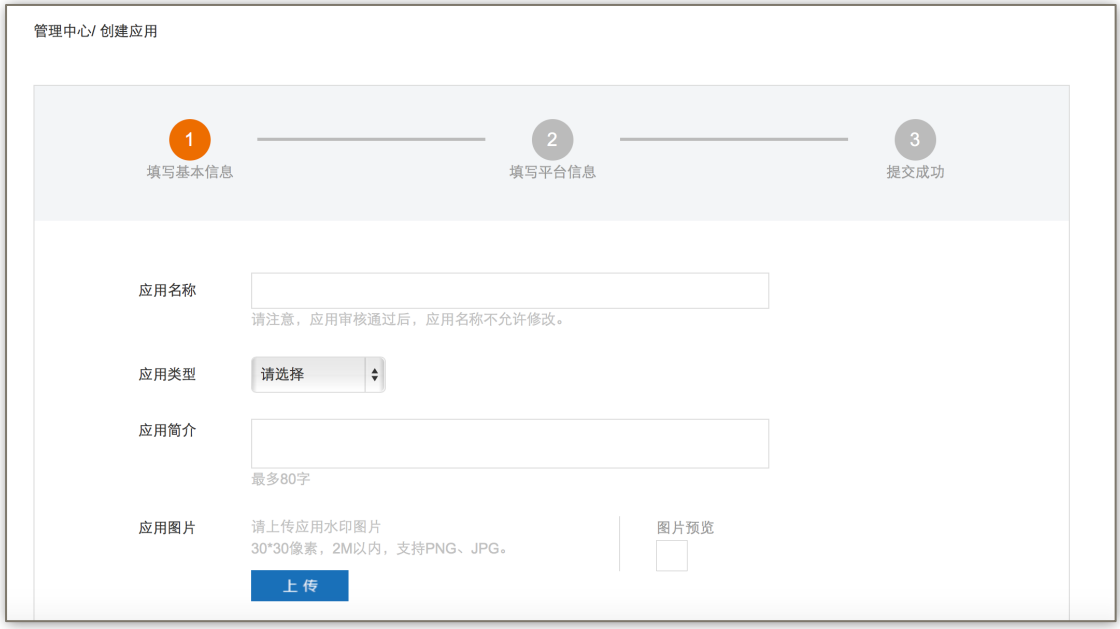
WO+通行证Android平台SDK平台

版本1.0.0

1.SDK接入设置

1)注册成为开发者

2)按要求填写基本信息



3)填写平台信息

2.SDK接入准备

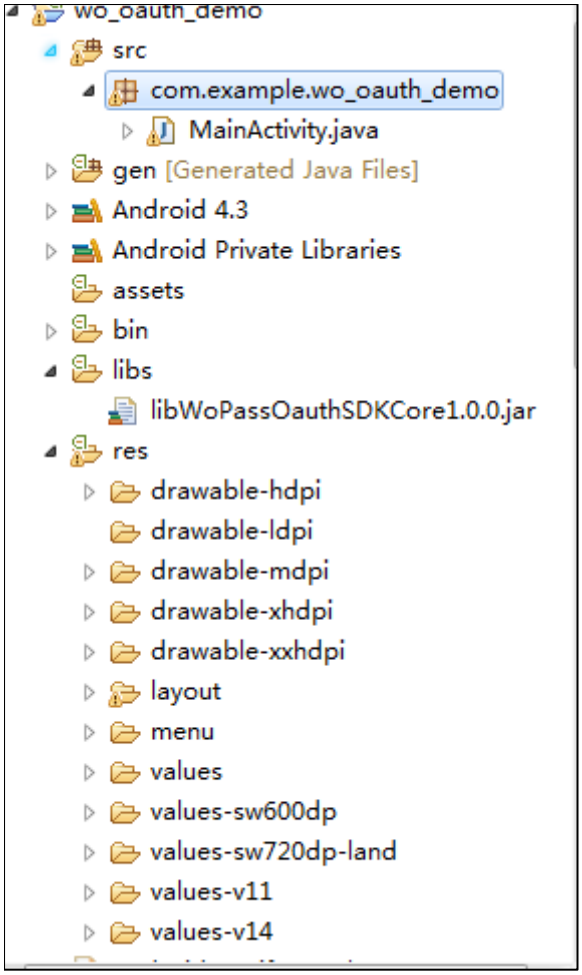
1)请在网站注册成为开发者，创建移动应用，获取client_id,client_secret信息。

2)添加SDK文件到工程

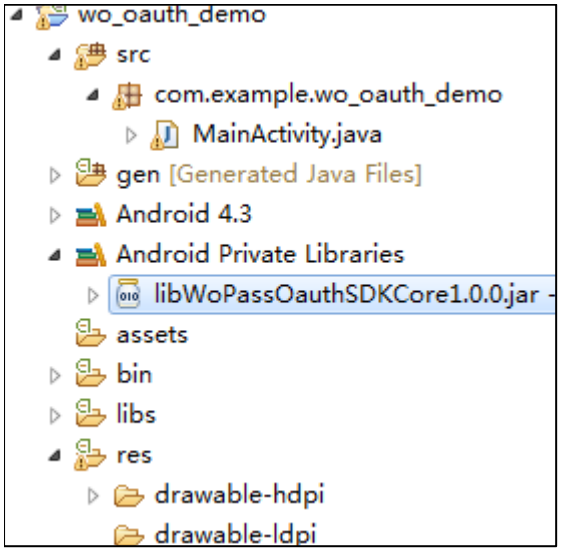
请在开放平台网站上下载libWoPassOauthSDKCore1.0.0.jar。

3.SDK接入流程：

(1)下载好的SDK文件添加到libs文件夹内（如下图所示：）



(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libWoPassOauthSDKCore1.0.0.jar文件。（如下图所示）。

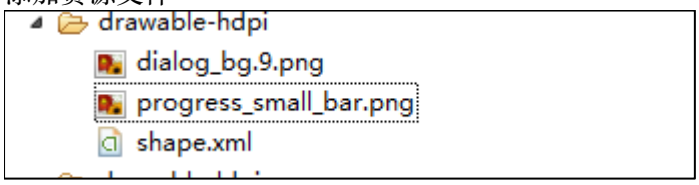


(3)在style中添加样式信息引入。

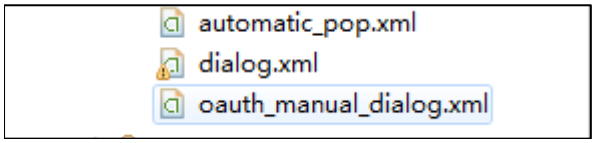
```xml
<style name="wo_passport_dialog" parent="@android:style/Theme.Dialog">
    <item name="android:windowFrame">@null</item>
    <item name="android:windowIsFloating">true</item>
    <item name="android:windowIsTranslucent">false</item>
    <item name="android:windowNoTitle">true</item>
    <item 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3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:

```xml
<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)引入clinet_id和client_secret与重定向url

```java
//用户申请的client_id
private String CLIENT_ID = "******";
//用户申请的client_secret
private String CLIENT_SECRET = "******";
// 重定向url
public static final String OAUTH_CALLBACK_URL = "******";
```

(6)实现回调接口，用于获取oauth登录的返回的数据

```java
implements WoOauthCallbackListener
```

实现这个接口的时候，可以实现两个方法，onResultCallback这个方法，用于返回请求数据，数据类型是String，onMongoliaLayerFinish 这个方法，是提供给开发者用于，取消蒙层使用的，当弹窗的时候回调这个方法。

自动模式:

本sdk，实现两类oauth登陆方式，第一种为自动模式，这个模式，可以延迟设定时间，弹

窗提醒用户进行登陆的，第三方需要判断用户是否登陆，如果未登录，可以启动这个模式（这个模式，会进行net取号，非联通号不能使用此功能，而且在wifi的情况下，因为不能进行net取号，也不能使用此功能）

(7)自动授权模式

自动授权模式，可以设置延迟时间，弹窗（如下图）提醒用户进行登陆的，第三方需要判断用户是否登陆，如果未登录，可以启动这个模式，这个会进行net取号，成功后，根据应用的级别如果，应用的级别是0：开发者直接拿到Access_token,token_type,openid等信息，应用级别是1的情况下，弹出如下的框，提示登陆，点击一键登陆，成功后，开发者拿到Access_token,token_type,openid等信息。

另外，根据第三方提供应用的安全级别，如果安全级别是0，则直接返回数据；如果安全级别是1，会弹窗如下图（安全级别是1）；如果安全级别是2的情况下，会弹出登陆页面，如下图（安全级别是2）

图： 安全级别1

图： 安全级别2

```
    /**
     * 延迟多长时间启动自动模式
     * @param delaytime  单位是秒
     */
    private void automaticAuthorizeMode(int delaytime) {
        float screenWidth = (float) (getWindowManager().getDefaultDisplay()
                .getWidth() * 0.8);
        float screenHeight = (float) (getWindowManager().getDefaultDisplay()
                .getHeight() * 0.5);
        WogarOAuth wogarOauth = new WogarOAuth(context);
        String state = wogarOauth.automaticAuthorizeMode(this, screenWidth,
                screenHeight, delaytime,R.style.wo_passport_dialog);
        tv_show.setText(state);//这里主要是自动模式，返回的状态
    }
```

参数说明：delaytime自动授权模式，延迟启动的时间，单位是秒。
返回数据会在回调函数中拿到，如下图：
(8)手动授权模式

```
private void manualAuthorizationMode() {
    WogarOAuth wogarOauth = new WogarOAuth(context);
    wogarOauth.setCodeParams(CLIENT_ID, CLIENT_SECRET, OAUTH_CALLBACK_URL);
    wogarOauth.manualAuthorizationMode(this,
            true,R.style.Dialog_Fullscreen);
}
```

manualAuthorizationMode这个方法中的，第2个参数，用于控制是否进行net取号，true的情况进行net取号，false的情况下不进行net取号，开发者根据业务需求，传入参数即可。
数据同样会在回调方法中获取，如access_token，openid等信息。

(9)获取用户信息

```
protected void getUserInfo() {
    new AsyncTask<Void, Void, String>() {
        @Override
        protected String doInBackground(Void... arg0) {
            // TODO Auto-generated method stub
            WogarOAuth wogarOauth = new WogarOAuth(context);
            String result = wogarOauth.getUserInfo(access_token, openid,
                    CLIENT_ID);
            return result;
        }

        @Override
        protected void onPostExecute(String result) {
            super.onPostExecute(result);
            tv_show.setText("..."+result);
        }
    }.execute();
}
```

参数说明：

CLIENT_ID:用户申请的client_id

CLIENT_SECRET:用户申请的client_secret

OAUTH_CALLBACK_URL:重定向地址

返回数据格式：

返回数据统一为json类型。

{"access_token":"07ec2967177f8a0fd3f96647aa9c6d20","token_type":"bearer","expires_in":"2564452","openid":"1531300156504"}