

# 新浪 SDK 接入说明

V2.1.2 版

时间	版本	更新说明
2014-3-7	V2.1.2	1. 增加登录、注册功能的流程图； 2. 增加登录成功后，游戏服务器验证登录 token 的流程； <b>2. 修改支付完成之后订单查询流程，将查询新浪服务器修改为查询游戏服务器；</b>
2014-2-26	V2.1.1	1. 优化了登录、注册流程、重新设计登录页面，增加用户安全性检测。 2. 增加多个新浪游戏之间用户数据共享； 3. 优化礼包、攻略页面的展示效果； 4. 优化了游戏加速功能； 5. 为游戏增加前后弹窗。
2013-12-9	V2.0.1	1. 用户登录成功之后增加浮动的小圆点； 2. 增加礼包卡，攻略、论坛、活动，存卡箱、消息中心等功能。 3. 修改支付接入代码，简化开发商接入流程；
2013-10-18	V1.0.1	1. 增加登录时切换账号按钮； 2. 修改切换账号时，出现崩溃的 BUG；
2013-10-9	V1.0	初稿

# 目录

一 接入前注意事项.....	4
二 导入并配置工程.....	4
1 导入 SINASDK 工程.....	4
2 配置游戏工程.....	5
1 添加权限.....	6
2 添加 Receiver、Provider、Service.....	7
3 添加 Activity.....	8
4 配置 AppKey、AppSecret、推送 Key.....	13
5 配置 intent-filter.....	13
6 配置 provider.....	14
7 配置游戏横竖屏.....	14
3 初始化 SDK.....	16
1 接口说明.....	17
2 初始化接入举例.....	18
三 登录接入.....	18
1 登录流程图.....	21
2 接口说明.....	21
3 登录过程举例.....	22
1 创建单例 Wyx.....	22
2 调用登录接口.....	22
3 重写 onActivityResult 函数.....	22
4 实现 AuthListener 接口.....	22
5 上传 token 验证并返回结果.....	23
四 游戏退出接入.....	24
1 接口说明.....	24
2 退出过程举例.....	24
1 创建单例 Wyx.....	24
2 调用退出接口.....	24
3 重写 onKeyDown() 方法.....	24
五 支付接入.....	25
1 支付流程图.....	25
2 接口说明.....	25
3 支付过程举例.....	26
1 创建单例 Wyx.....	26
2 调用下订单接口.....	26
3 实现下订单回调接口.....	26
4 实现支付窗口关闭回调接口.....	27
5 支付完成后跳转回到游戏，支付完成.....	27
6 回调过程.....	27
7 支付常见问题.....	28
六 打渠道包.....	28
七 附录.....	30

1 新浪 TOKEN 验证接口 ..... 30

## 一 接入前注意事项

如果当前游戏是**单机游戏**，packageName 需要以”**\_sinas**”结尾；如果当前游戏是**网游**，packageName 需要以”**\_wyx**”结尾，如下图所示：

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.testgame_wyx"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="8"
9         android:targetSdkVersion="17" />
10
```

图 1 网游的 packageName

SinaSDK 工程中，已经自带了 `android-support-v4.jar`，因此开发商的游戏工程中就不必再重复引入该 jar 包，否则会发生错误。

## 二 导入并配置工程

### 1 导入 SinaSDK 工程

在 Eclipse 中导入 SinaSDK 工程，并且将该工程指定为一个 Library，设置工程编码为：UTF-8，如下图：

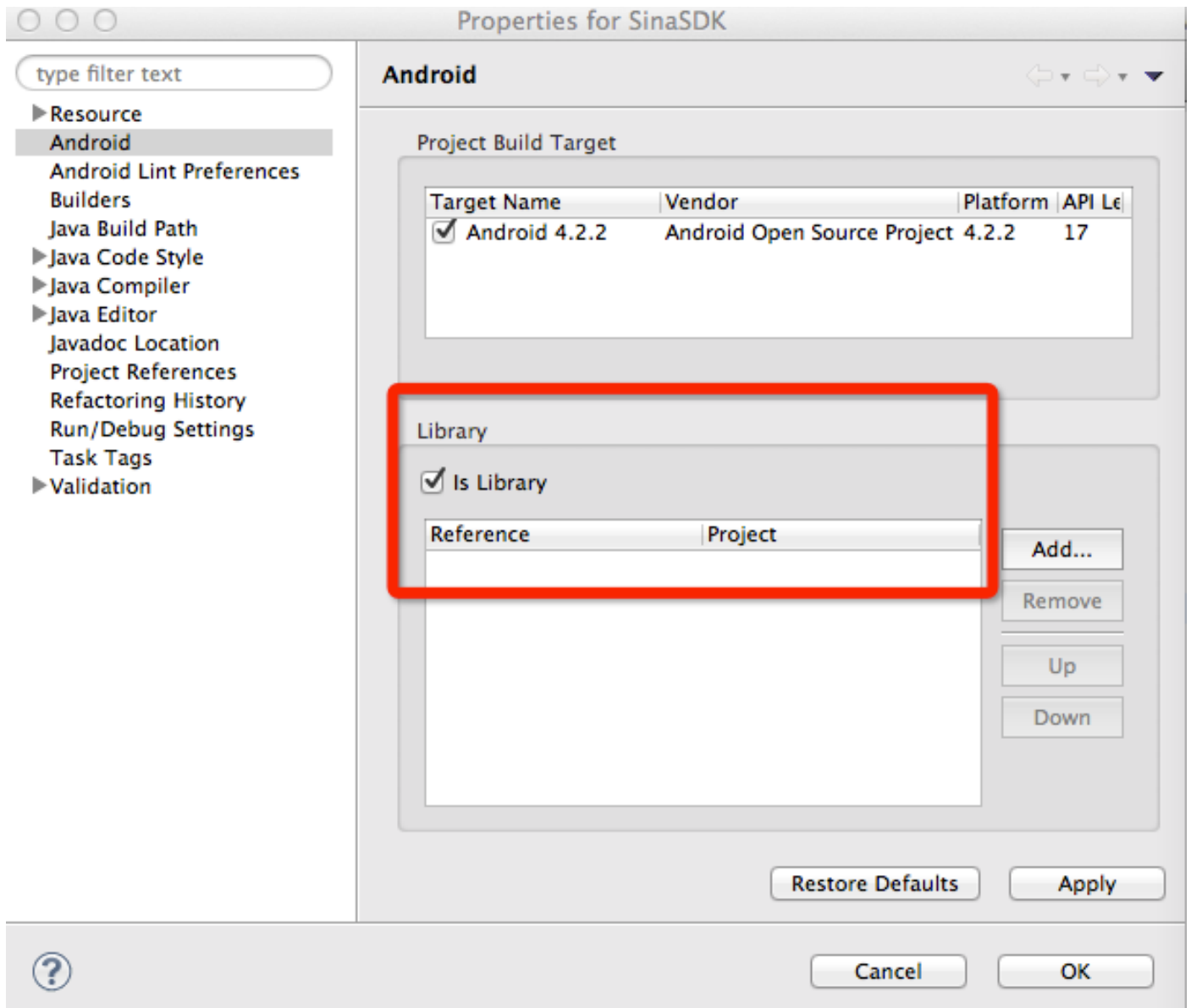


图 2 导入 SinaSDK 工程

## 2 配置游戏工程

打开需要接入新浪 SDK 的游戏工程，引用 SinaSDK 工程，如下图所示：

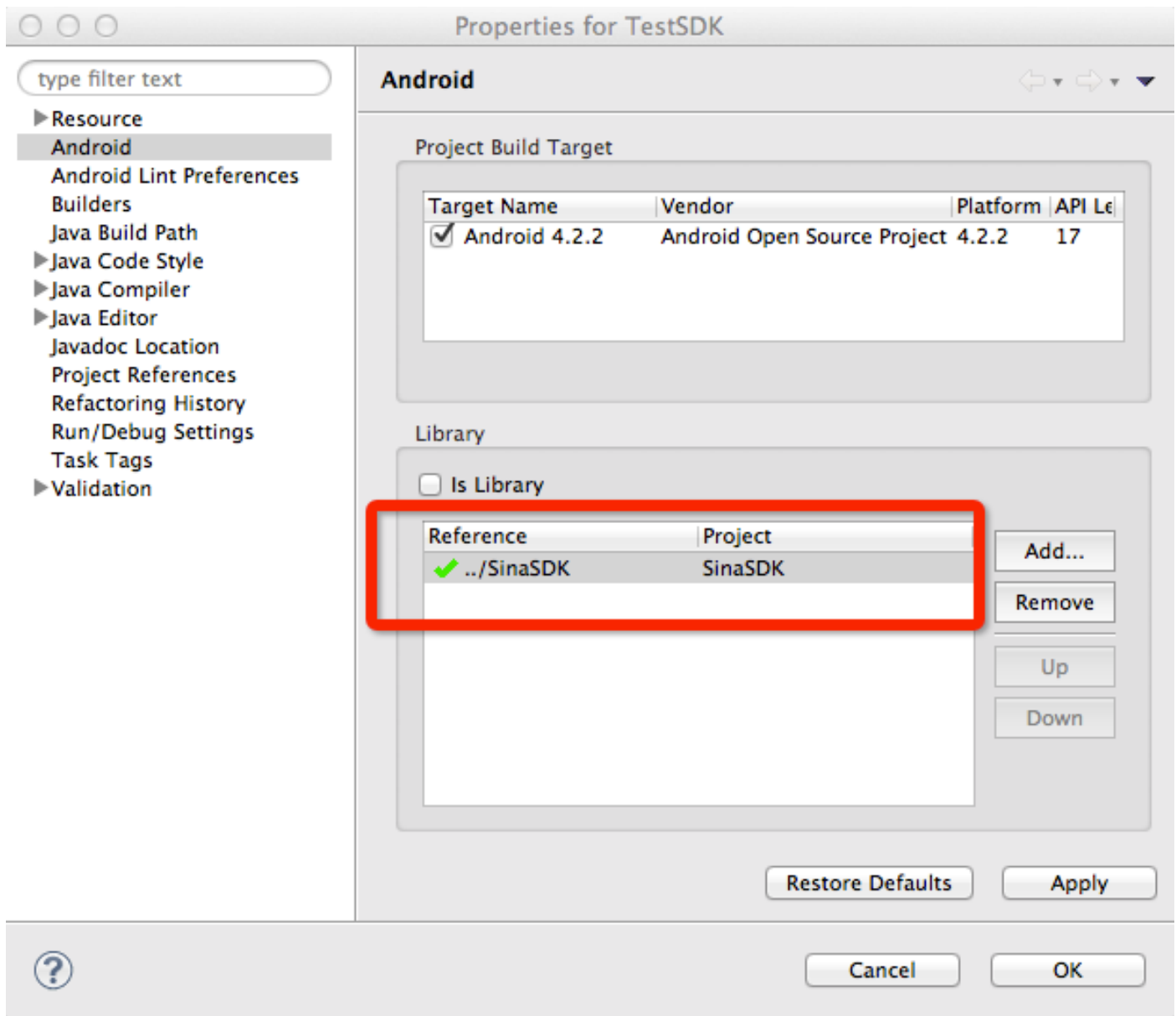


图 3 增加对 SinaSDK 工程的引用

## 1 添加权限

在游戏工程 AndroidManifest.xml 中，增加以下权限，请添加到 `<manifest>` 节点下，（可参考 TestSDK 里面的 AndroidManifest.xml）：

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.VIBRATE" />
```

```

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.RECEIVE_USER_PRESENT" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.RESTART_PACKAGES" />
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"
/>

```

## 2 添加 Receiver、Provider、Service

在游戏工程 AndroidManifest.xml 中，增加以下内容，请添加到 <application 节点下，（可参考 TestSDK 里面的 AndroidManifest.xml）：

```

<!-- 4个receiver -->
<receiver android:name="com.igexin.sdk.SdkReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
        <action android:name="android.intent.action.USER_PRESENT" />
    </intent-filter>
</receiver>
<receiver
android:name="com.igexin.sdk.coordinator.NotificationCenterAIDLReceiver" >
    <intent-filter>
        <action android:name="com.igexin.sdk.action.refreshls" />
    </intent-filter>
</receiver>
<receiver
    android:name="sdk.download.DownloadReceiver"
    android:exported="false" >
    <intent-filter>
        <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
    </intent-filter>
</receiver>
<receiver
    android:name="cn.sina.youxi.app.push.GexinSdkMsgReceiver"
    android:exported="false" >
    <intent-filter>
        <action
android:name="com.igexin.sdk.action.cLCYAFri5X6Kzha4pLbcb6" />
    </intent-filter>
</receiver>

```

```

<!--
    1个provider
    注: android:authorities中, "sdk.download." 后面的部分, 一定要配置为当前游戏的
packageName
    否则会和其他游戏冲突, 导致用户无法安装
    例如:
    当前packageName为: com.example.xxxx, 则
    android:authorities="sdk.download.com.example.xxxx"
-->
<provider
    android:name="sdk.download.DownloadProvider"
    android:authorities="sdk.download.com.example.testgame_wyx"
    android:process=":pushservice" />

<!-- 3个service -->
<service
    android:name="com.igexin.sdk.coordinator.SdkMsgService"
    android:exported="true"
    android:label="NotifyCenterAIDL"
    android:process=":pushservice" >
</service>
<service
    android:name="sdk.download.DownloadService"
    android:process=":pushservice" />
<service
    android:name="com.igexin.sdk.SdkMainService"
    android:exported="false"
    android:label="NotifyCenter"
    android:process=":pushservice" >
</service>

```

### 3 添加 Activity

在游戏工程 AndroidManifest.xml 中, 添加以下 Activity, 请添加到<application 节点下, (可参考 TestSDK 里面的 AndroidManifest.xml) :

```

<!-- sdk activity,begin -->
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountManageActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
    android:launchMode="singleTop"
    android:theme="@style/gamehall_account_dialog"
    android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity

```



```

        android:name="cn.sina.youxi.pay.sdk.ui.AccountLoginingActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@android:style/Theme.Translucent"
        android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountLoginVerifyActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@style/gamehall_account_dialog"
        android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountRegisterActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@android:style/Theme.Light.NoTitleBar"
        android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountRegisterVerifyActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@android:style/Theme.Light.NoTitleBar"
        android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountRegisterSuccActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@style/gamehall_account_dialog"
        android:windowSoftInputMode="stateAlwaysHidden" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountResetActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@android:style/Theme.Light.NoTitleBar" >
</activity>
<activity
    android:name="cn.sina.youxi.pay.sdk.ui.AccountResetPhoneActivity"

    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:theme="@android:style/Theme.Light.NoTitleBar" >
</activity>

```

```

        <activity
            android:name="cn.sina.youxi.pay.sdk.ui.AccountResetEmailActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@android:style/Theme.Light.NoTitleBar" >
        </activity>
        <activity
            android:name="cn.sina.youxi.pay.sdk.ui.AccountBindingActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@android:style/Theme.Light.NoTitleBar" >
        </activity>
        <activity
            android:name="cn.sina.youxi.pay.sdk.ui.AccountSecurityCenterActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@android:style/Theme.Light.NoTitleBar" >
        </activity>
        <activity
            android:name="cn.sina.youxi.app.push.ADAActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@android:style/Theme.Dialog" >
        </activity>
        <activity
            android:name="com.igexin.sdk.SdkActivity"
            android:excludeFromRecents="true"
            android:process=":pushservice"
            android:taskAffinity="android.task.SdkActivityTask"
            android:theme="@android:style/Theme.Translucent.NoTitleBar" >
            <intent-filter>
                <action
android:name="com.igexin.action.popupact.com.example.testgame_wyx" />

                    <category android:name="android.intent.category.DEFAULT" />
                </intent-filter>
            </activity>
        <activity
            android:name="cn.sina.youxi.app.FloatMenuActivity"

android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:launchMode="singleTop"
            android:theme="@style/gamehall_transparent"
            android:windowSoftInputMode="stateAlwaysHidden" >
        </activity>
        <activity
            android:name="cn.sina.youxi.app.PopupDetailsView"

```

```

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent"
        android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.MessageCenterActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent"
        android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.app.UniversalWebviewActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent"
        android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.SystemMessageActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent"
        android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.FeedBackFragmentActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent"
        android:windowSoftInputMode="stateAlwaysHidden|adjustResize" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.CardFragmentActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
        android:launchMode="singleTop"
        android:theme="@style/gamehall_transparent" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.FeedbackRecord"

```

```

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:launchMode="singleTop"
            android:theme="@style/gamehall_transparent"
            android:windowSoftInputMode="stateAlwaysHidden|adjustResize" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.GuideDetailActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:launchMode="singleTop"
            android:theme="@style/gamehall_transparent" >
    </activity>
    <activity
        android:name="cn.sina.youxi.pay.sdk.ui.AccountRegisterErrorActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@style/gamehall_account_dialog"
            android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.ActDetailActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@style/gamehall_transparent"
            android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.GiftDetailActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@style/gamehall_transparent"
            android:windowSoftInputMode="stateAlwaysHidden"
            android:hardwareAccelerated="false">
    </activity>
    <activity
        android:name="cn.sina.youxi.ui.FeedbackSendActivity"

        android:configChanges="orientation|keyboardHidden|navigation|screenSize"
            android:theme="@style/gamehall_transparent"
            android:windowSoftInputMode="stateAlwaysHidden" >
    </activity>
    <!-- sdk activity,end -->

```

## 4 配置 AppKey、AppSecret、推送 Key

在游戏工程 AndroidManifest.xml 中配置 AppKey、AppSecret (android:values 中配置的值, 需要有 WYX\_前缀)、推送 Key, 添加在<application 结点下(可参考 TestSDK 里面的 AndroidManifest.xml), 如下:

```
<!-- 推送相关配置 请勿修改 -->
<meta-data
    android:name="appid"
    android:value="cLCYAFri5X6Kzha4pLbcb6" />
<meta-data
    android:name="appkey"
    android:value="xlIMijq4KVATbRns0Kyxf" />
<meta-data
    android:name="appsecret"
    android:value="wGLck6gTxeAfJBMLSEsI05" />
<!-- 推送相关配置 请勿修改 -->

<!-- 此处配置AppKey, 不要去掉android:value中的前缀: WYX_ -->
<meta-data
    android:name="WYX_APPKEY"
    android:value="WYX_3093918086" >
</meta-data>

<!-- 此处配置AppSecret, 不要去掉android:value中的前缀: WYX_ -->
<meta-data
    android:name="WYX_APPSECRET"
    android:value="WYX_920859f7bc730cccedfafc20e657d187" >
</meta-data>
```

其中, 推送 Key 无需修改, WYX\_APPKEY、WYX\_APPSECRET 需要配置成当前游戏申请到的值。

## 5 配置 intent-filter

### 第 1 步

找到游戏本身的入口 Activity, 增加以下 intent-filter:

```
<intent-filter>

    <action android:name="android.intent.action.VIEW" />

    <category android:name="android.intent.category.DEFAULT" />
```

```

<category android:name="android.intent.category.BROWSABLE" />

    <data android:host="mainpage"

        android:scheme="wyx3093918086" />

</intent-filter>

```

其中，该 filter 中的 android:scheme 属性中的“3093918086”是游戏的 appKey，请照实填写。

## 第 2 步

找到 android:name 为“com.igexin.sdk.SdkActivity”的 Activity，修改其 filter 为以下内容：

```

<intent-filter>
    <action
        android:name="com.igexin.action.popupact.com.example.testgame_wyx" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>

```

其中，该 filter 中 android:name 属性中的“com.example.testgame\_wyx”是游戏的 packageName，请照实填写。

## 6 配置 provider

找到 android:name=”sdk.download.DownloadProvider”的 provider，修改 android:authorities 为：“sdk.download.com.example.testgame\_wyx”，其中“com.example.testgame\_wyx”是当前游戏的 packageName，请照实填写，如下图所示：

```

69
70  --
71      1个provider
72      注：android:authorities中，“sdk.download.”后面的部分，一定要配置为当前游戏的packageName
73      否则会和其他游戏冲突，导致用户无法安装
74      例如：
75      当前packageName为：com.example.xxxx，则android:authorities="sdk.download.com.example.xxxx"
76  -->
77      <provider
78          android:name="sdk.download.DownloadProvider"
79          android:authorities="sdk.download.com.example.testgame_wyx"
80          android:process=":pushservice" />
81

```



**请注意，这一步非常重要，如果配置错误，会与其他游戏产生冲突，导致用户不能安装！！**

## 7 配置游戏横竖屏

如果当前游戏是竖屏游戏，则需要在游戏的 AndroidManifest.xml 中，对调用登录接口的

Activity，进行如下配置。比如，你在 GameActivity 中调用了登录接口，就应该对 GameActivity 进行配置：

```
android:screenOrientation="portrait"
```

如下图所示：

A screenshot of an AndroidManifest.xml file. The <activity> tag is expanded, showing various attributes. The attribute android:screenOrientation="portrait" is highlighted with a red rectangular box. Other visible attributes include android:name="com.example.testgame\_wyx.GameActivity", android:configChanges="orientation|keyboardHidden|navigation|screenSize", and android:label="@string/app\_name". The intent-filter section contains actions for MAIN, VIEW, and categories for LAUNCHER, DEFAULT, and BROWSABLE. A data element specifies host="mainpage" and scheme="wyx3093918086".

```
<activity
    android:name="com.example.testgame_wyx.GameActivity"
    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
    android:label="@string/app_name"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data
            android:host="mainpage"
            android:scheme="wyx3093918086" />
    </intent-filter>
</activity>
```

图 4 竖屏游戏 Activity 配置

如果当前游戏是横屏游戏，则进行如下配置：

```
android:screenOrientation="landscape"
```

如下图所示：

```

<activity
    android:name="com.example.testgame_wyx.GameActivity"
    android:configChanges="orientation|keyboardHidden|navigation|screenSize"
    android:label="@string/app_name"
    android:screenOrientation="landscape" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />

        <data
            android:host="mainpage"
            android:scheme="wyx3093918086" />
        </intent-filter>
    </activity>

```

图 5 横屏游戏 Activity 配置

### 3 初始化 SDK

开发商在进入游戏之后，调用登录、支付等接口之前，必须先调用 SDK 初始化方法（本接口只需要调用一次，不需要重复调用）。

例如，可以在游戏启动页 Activity 的 onCreate() 方法中调用，在调用初始化方法的 Activity 上方，会出现如下图所示的加速条：





## 1 接口说明

```
/**
 * 初始化配置
 *
 * @param instance
 */
public void initConfig(final Activity instance)
```

## 2 初始化接入举例

### 1 创建单例 Wyx

```
mWyx = Wyx.getInstance(mContext);
```

### 2 调用初始化接口

```
mWyx.initConfig(this);
```

## 三 登录接入

开发商要使用新浪账户体系登录游戏，必须调用登录接口。通常在游戏主页的 `onCreate()` 方法中调用。

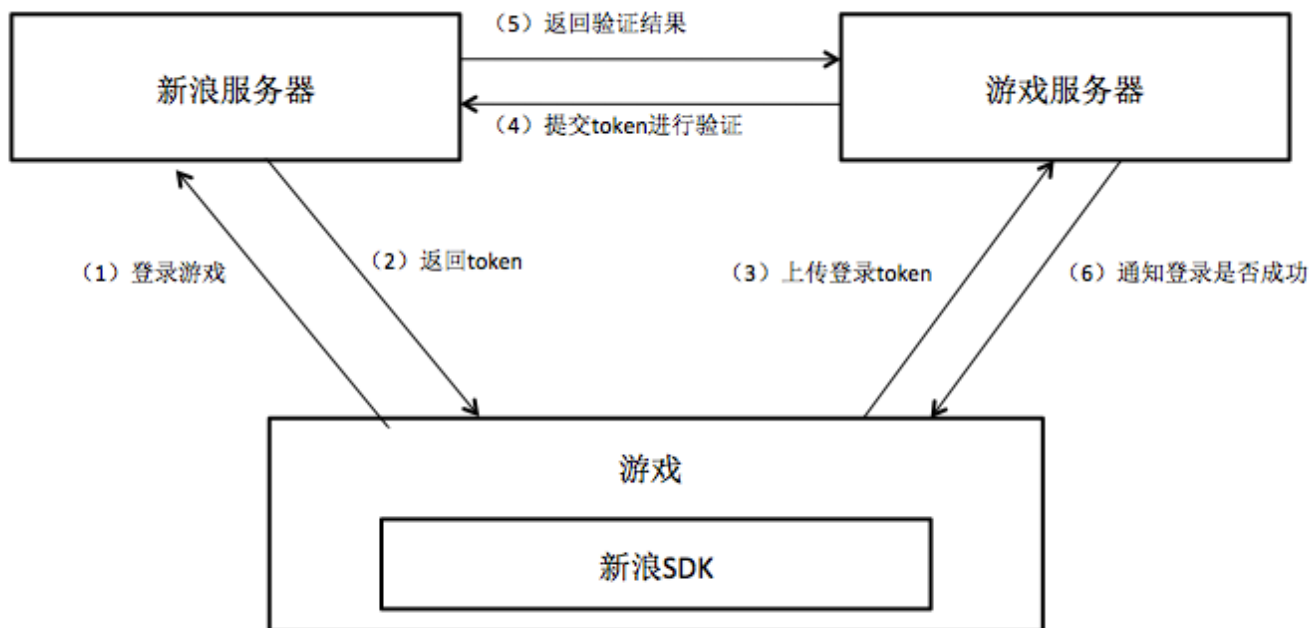
如果之前未登录过，则会弹出如下图所示窗口：



如果当前已经登录过，则会弹出如下图所示窗口：



## 1 登录流程图



### 说明:

开发商调用新浪登录接口成功登录之后，可以获取到新浪返回的 token，开发商需要自行将该 token 传给自己的游戏服务器（第 3 步），游戏服务器收到客户端传来的 token，需要调用新浪服务器的 token 验证接口来验证该 token 是否合法（第 4 步），验证 token 的接口，请参照附录中的详细说明。

## 2 接口说明

```
/**
 * 登录
 * 如果当前未登录过，则跳转到登录页面
 * 如果当前已经登录过，则校验本地存储的token、uid是否合法，不合法则重新登录
 *
 * @param activity
 * @param wyxAuthListener
 */
public void authorize(Activity activity, WyxAuthListener wyxAuthListener)

/**
 * 注销当前用户登录
 */
```

```
public void logout()
```

### 3 登录过程举例

#### 1 创建单例 Wyx

```
mWyx = Wyx.getInstance(mContext);
```

#### 2 调用登录接口

```
mWyx.authorize(this, mAuthListener);
```

#### 3 重写 onActivityResult 函数

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);

    if (mWyx != null) {
        mWyx.authorizeCallBack(requestCode, resultCode, data);
    }
}
```

#### 4 实现 AuthListener 接口

必须实现 AuthListener 接口,并在 AuthListener 接口的 onComplete() 回调方法中调用以下方法:

```
mWyx.initFloatView(GameActivity.this, values);
```

代码实例:

```
class AuthDialogListener implements WyxAuthListener {

    @Override
    public void onComplete(Bundle values) {

        //必须初始化浮动窗口

        mWyx.initFloatView(GameActivity.this, values);

        // TODO 在此处写回调逻辑

        setUserInfo();
    }
}
```

```

String toast = "(测试信息)：登录过程完成，结果：";

boolean isLogin = values != null ? values.getBoolean("isLogin")
    : false;

if (isLogin && mWyx.isLogin(mContext)) {
    toast += "成功，" + mWyx.getUserName();
} else {
    toast += "失败";
}

Toast.makeText(GameActivity.this, toast, Toast.LENGTH_SHORT).show();
}

@Override

public void onFail(Exception e) {
    e.printStackTrace();
}

@Override

public void onCancel() {
    // TODO 测试信息

    Toast.makeText(GameActivity.this, "(测试信息)：登录过程取消",
        Toast.LENGTH_SHORT).show();
}
}

```

## 5 上传 token 验证并返回结果

对应流程图中第 3 步，开发商需要把获取到的新浪 token，以及 uid、MD5 后的手机串号、ip 地址，上传至游戏服务器。

对应流程图中第 4、5 步，开发商需要在服务器端调用新浪的 token 验证接口，验证当前客户端上传的 token 是否有效。

对应流程图中第 6 步，游戏服务器将 token 的验证结果返回给客户端，客户端提示用户登录成功/失败。

新浪 token、uid、MD5 后的手机串号、ip 地址等参数的获取方式，请参照附录中新浪 token 验证接口的说明。

详细的登录示例，请参考 TestSDK 工程中 GameActivity。

## 四 游戏退出接入

### 1 接口说明

```
/**
 * 游戏退出
 *
 * @param instance
 * @return
 */
public void destroy(final Activity instance)
```

### 2 退出过程举例

#### 1 创建单例 Wyx

```
mWyx = Wyx.getInstance(mContext);
```

#### 2 调用退出接口

游戏退出时，必须调用：

```
mWyx.destroy(mInstance);
```

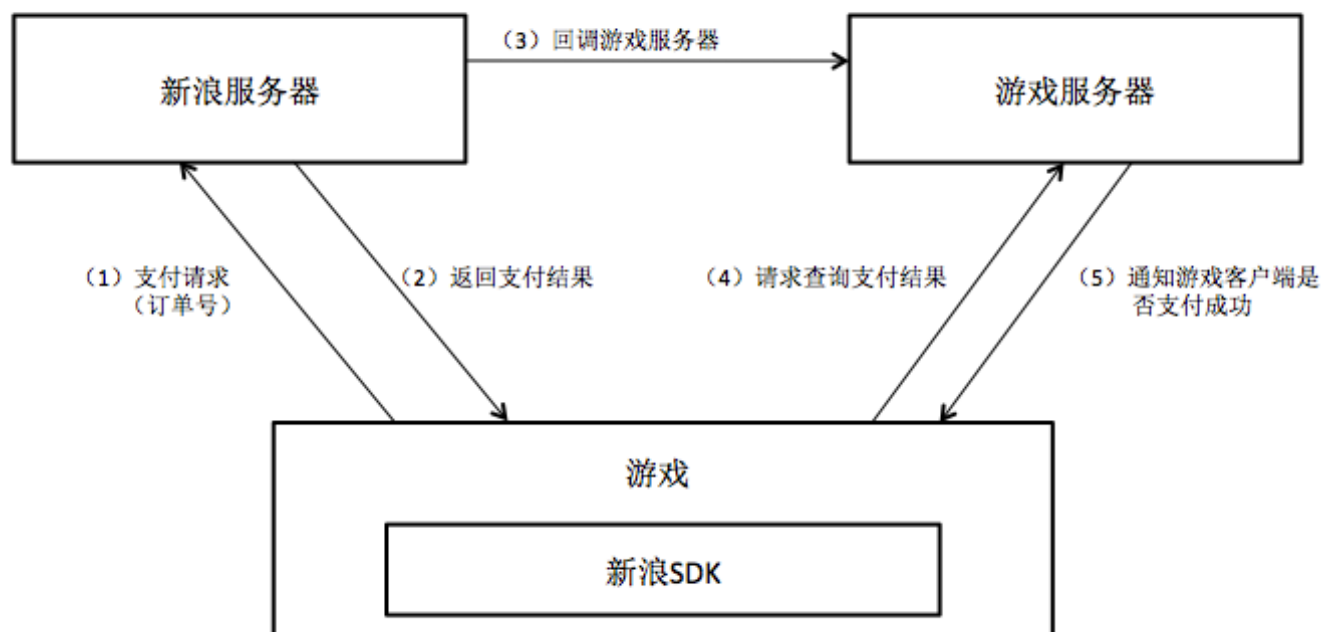
#### 3 重写 onKeyDown()方法

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK) {
        mWyx.destroy(GameActivity.this);
        return true;
    } else {
        return super.onKeyDown(keyCode, event);
    }
}
```



# 五 支付接入

## 1 支付流程图



### 说明：

由于网络请求的不确定性，用户发起查询（第 4 步）的时候，回调过程（第 3 步）可能尚未结束，因此开发商的需要发起多次查询（第 4 步），直到查询到当前订单状态。例如，开发商可以维护一个定时器，定时到游戏服务器上查询。

## 2 接口说明

/\*\*

- \* 下订单（带透传参数），成功之后打开支付页面 透传参数从客户端传给新浪服务器，
  - \* 新浪服务器在回调的时候带上该参数回调游戏服务器
  - \* 新浪调用游戏服务器的参数名为"pt"
  - \*
  - \* @param instance 当前Activity实例
  - \* @param amount 金额，单位为分
  - \* @param desc 商品描述，将会显示在支付页面，不能为空
  - \* @param customParam 透传参数，传入空值则说明无透传参数
  - \* @param responseListener 下订单的回调接口
  - \* @param dismissListener 支付页面关闭时的回调，如想查询订单状态，可以在该回调中实现
- \*/

```
public void placeOrder(final Activity instance, long amount, String desc,
    String customParam, final ResponseListener responseListener,
```

```

        final OnDismissListener dismissListener)

/**
 * 订单查询接口
 * @param order_id 订单号
 * @param requestListener
 */
public void queryOrder(String order_id, ResponseListener requestListener)

```

## 3 支付过程举例

### 1 创建单例 Wyx

```
mWyx = Wyx.getInstance(mContext);
```

### 2 调用下订单接口

```

mWyx.placeOrder(PayDemoActivity.this, 1, "一次霸气的测试",
                "自定义参数", mPlaceOrderListener,
                mPayDialogDismissListener);

```

### 3 实现下订单回调接口

```

//必须实现下订单回调接口
class PlaceOrderListener implements ResponseListener {

    @Override
    public void onComplete(String response) {
        //下订单完成之后会立即回调接口

        if (mProgressDialog.isShowing()) {
            mProgressDialog.dismiss();
        }

        JSONObject jsonObj = JSONUtils
            .parse2JSONObject(response);
        String code = JSONUtils.getString(jsonObj, "code");

        if (!TextUtils.isEmpty(code)) {
            Log.e(TAG, "placeOrder, 下订单过程中报错: " + response);
        }
        else {
            mOrderId= JSONUtils.getString(
                jsonObj, "order_id");

```

```

        Log.i(TAG, "placeOrder, 下订单之后获取到订单号: "+ mOrderId);
    }
}

@Override
public void onFail(Exception e) {
    if (mProgressDialog.isShowing()) {
        mProgressDialog.dismiss();
    }
}
}

```

## 4 实现支付窗口关闭回调接口

// 支付窗口关闭时会回调该接口，如果想查询订单状态，可以在该回调中实现

```
class PayDialogDismissListener implements OnDismissListener {
```

```

    @Override
    public void onDismiss(DialogInterface dialog) {
        if (mProgressDialog.isShowing()) {
            mProgressDialog.dismiss();
        }

        // TODO
        Log.i(TAG, "支付窗口关掉，当前订单号: " + mOrderId);
        Toast.makeText(PayDemoActivity.this, "(测试信息)：支付窗口关掉，当前订单号: "
            + mOrderId, Toast.LENGTH_SHORT)
            .show();

        // TODO
        // 此处查询游戏服务器，获取当前订单状态
    }
}

```

## 5 支付完成后跳转回到游戏，支付完成

这一步跳转由 SDK 来完成，开发商不需要做任何事情，支付过程可能成功，也可能失败。

## 6 回调过程

支付完成之后，新浪支付服务器会立即回调开发商的回调地址，通知开发商支付结果。

这一步都是在服务器完成，客户端不需要做任何事情，但是需要开发商在游戏服务器上提供一个

回调地址，回调地址的参数说明，请参照附件 [《附录一：手机游戏统一支付接口说明.doc》](#) 中关于回调接口部分的说明。

另外，现在的 SDK 也支持透传参数功能，开发商可以将参数通过游戏客户端传给新浪的服务器，新浪服务器在回调时，会带上这个参数。具体透传参数的使用可以参照 TestSDK 中支付的例子。

例如，开发商在支付的时候，可以将当前用户的角色 ID 当做一个透传参数传递给新浪服务器，新浪服务器在回调游戏服务器的回调接口时，会原样的带上这个参数。

## 7 支付常见问题

### 问题一：新浪 token 验证问题

#### 游戏服务器怎么才能校验客户端传来的 token 是否合法？

请参考附录中关于新浪 token 验证接口的说明

### 问题二：订单状态查询

#### 客户端调用查询订单接口，应该在什么时机去调用？

开发商可以在支付页面关闭的时候可以发起查询，在以上提到的 OnDismissListener 接口的回调方法实现。

开发商也可以在游戏里面的某些点，比如进入游戏购买道具画面，或者刚刚进入游戏的时候，主动查询开发商的服务器看看是否道具支付成功以决定是否更新画面道具信息。

由于网络请求的不确定性，用户发起查询的时候，回调过程可能尚未结束，因此开发商的需要发起多次查询，直到查询到当前订单状态。例如，开发商可以维护一个定时器，定时到游戏服务器上查询。

## 六 打渠道包

渠道号配置在游戏工程的 `assets\sina\configv2.properties` 文件中。

例如，当前游戏包的渠道号是 100010041000，则需要在游戏工程的该文件中，写入以下内容：

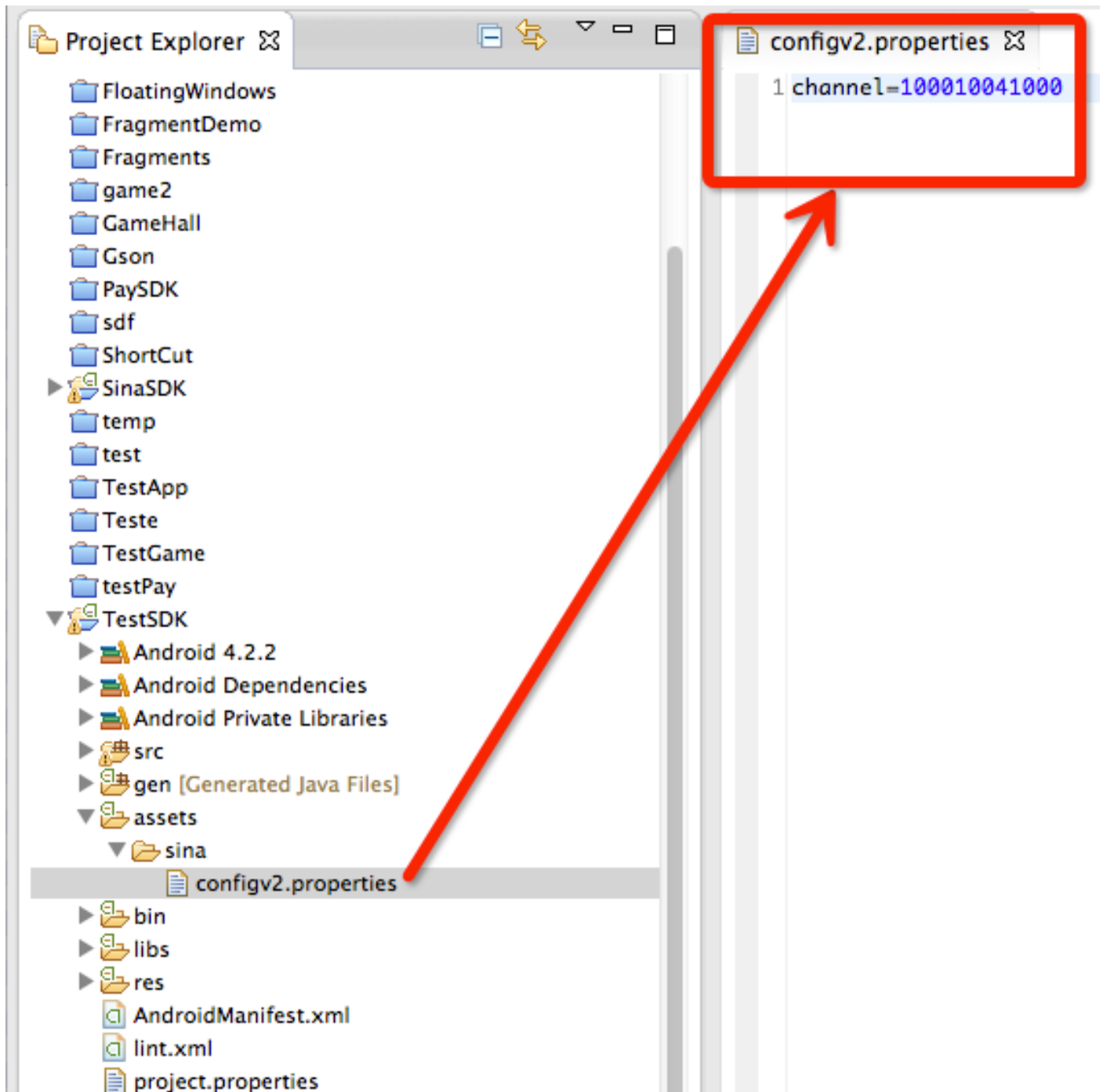


图 6 渠道号配置

新接入游戏只需要提供一个渠道值为 100010041000 的渠道包。

已接入的老游戏更新客户端，需要根据初次提供的渠道值对应打包，当时打过 N 个渠道包，这次依然需要打 N 个。

# 七 附录

## 1 新浪 token 验证接口

新浪提供了一个 token 校验接口，通过调用该接口可以校验 token 是否有效。

游戏客户端可以在支付的时候，同时把自己把获得的 token、uid、MD5 后的手机串号、ip 地址，发送到游戏服务器，服务器调用以下接口，如果返回结果成功，则说明当前 uid、token 合法。

客户端需要手动上传的参数可以使用以下方式获取：

token 的获取方式：`Wyx.getInstance(mContext).getToken();`

uid 的获取方式：`Wyx.getInstance(mContext).getUserId();`

MD5 后的手机串号的获取方式：

`String machineid = Wyx.getInstance(mContext).getEncryptIMEI();`

ip 地址的获取方式：`String ip = Wyx.getInstance(mContext).getIPAddress();`

**请务必使用以上方式来获取。**

token 校验接口地址：

`http://game.weibo.cn/sso.php`

调用方式：

get/post

参数说明：

参数名	说明	类型	描述
i	接口名称，请传入“chktoken”	String	请传入“chktoken”
uid	当前用户 uid	String	客户端传来的 uid
token	当前用户 token	String	客户端传来的 token
machineid	当前手机串号	String	客户端传来的 MD5 加密后的手机串号
ip	当前手机 IP 地址	String	客户端传来的 ip 地址

m	校验值	String	Md5(token+p) 其中 p 为固定字符串： String p = “d940a1ab0429ce60292187d9d70bae34” ;
---	-----	--------	--

返回值:

成功:

```
{
    "result": "succ",
    "data": {
        "uid": "001345",
        "token": "XXXXXX"
    }
}
```

失败:

```
{
    "result": "fail",
    "data": {
        "code": "001",
        "message": "error message"
    }
}
```

示例代码:

```
public void checkToken(String uid, String token, String machineid, String ip,
    ResponseListener responseListener) {

    String SSO_M = "d940a1ab0429ce60292187d9d70bae34";
    String POST_URL = "http://game.weibo.cn/sso.php";

    Bundle params = new Bundle();
    params.putString("i", "chktoken");
    params.putString("uid", uid);
    params.putString("token", token);
    params.putString("machineid", machineid);
    params.putString("ip", ip);

    // Md5(token+p)
    String source = token + SSO_M;
    params.putString("m", MD5Util.encrypt(source));

    mWyxAsyncRunner.request(POST_URL, params, "POST",
```

```
        responseListener);  
    }
```