

百度移动游戏 SDK For Android 参考手册

V3.7.4

2017-03-09

时间	版本	更新说明
2014-11-03	3.0.0	
2015-01-15	3.1.2	<ol style="list-style-type: none"> 1. 去掉版本渠道配置 2. 增加兼容多酷手游 SDK 的 APPID 和 APPKEY 设置
2015-04-27	3.2.0	<ol style="list-style-type: none"> 1. 增加显示和关闭悬浮窗接口 2. 修改其他须接入的 API
2015-06-30	3.2.2	<ol style="list-style-type: none"> 1. 增加获取公告接口
2015-09-15	3.5.0	<ol style="list-style-type: none"> 1. 修改 Androidmanifest.xml 的 QQ 钱包配置
2015-10-21	3.5.2	<ol style="list-style-type: none"> 1. 固件版本最低支持 2.3
2015-12-02	3.6.0	<ol style="list-style-type: none"> 1. Androidmanifest.xml 增加微信支付配置 2. 增加工程 assets 配置说明
2016-01-21	3.6.2	<ol style="list-style-type: none"> 1. Androidmanifest.xml 配置调整 2. 增加 onResume, onPause, setSupportScreenRecord, isSupportScreenRecord gameExit 等接口。 3. 取消 destroy 接口
2016-04-08	3.7.0	<ol style="list-style-type: none"> 1. 将 BDGameApplication 声明为放弃 2. 调整 Application 配置 3. 不再提供 setSupportScreenRecord 方法
2016-05-10	3.7.1	增加 查询登录用户实名认证状态接口
2016-07-10	3.7.2	剥离录屏 SDK，去掉相关的混淆
2016-10-21	3.7.3	<ol style="list-style-type: none"> 1. 支付接口增加 uid 验证。 2. 增加自定义统计接口。
2016-03-09	3.7.4	<ol style="list-style-type: none"> 1. 修复相关已知 bug 2. 优化渠道统计功能

目录

百度移动游戏 SDK For Android 参考手册1

1、 接入准备.....5

1.1 环境.....5

1.2 参数.....6

1.3 附加信息.....6

2、 SDK 配置.....6

2.1 Application 配置.....6

2.2 SDK 接入类型配置.....7

2.3 Permission、Activity、Service 等相关配置7

2.4 assets 配置.....10

2.5 debug 模式.....11

3、 API 接入说明11

3.1 API 接入顺序.....11

3.2 必须接入的 API 接口.....12

3.3 API 回调接口说明.....12

4、 API 功能介绍13

4.1 初始化（必接）13

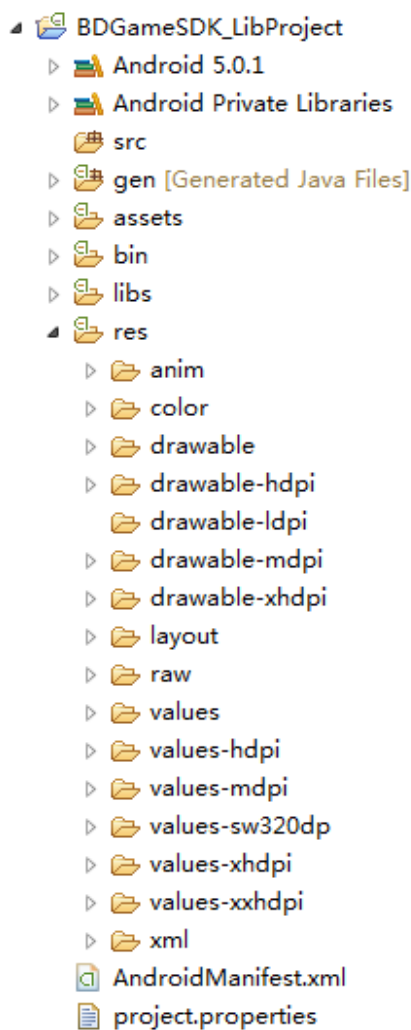
4.2 登录（必接）14

4.3	是否已登录（选接）	15
4.4	登录用户 UID（必接）	15
4.5	登录用户 AccessToken（必接）	16
4.6	注销（选接）	16
4.7	支付（必接）	16
4.8	设置切换账号结果通知（必接）	20
4.9	设置会话失效监听（必接）	21
4.10	设置多酷 AppId 和 AppKey（兼容多酷手游 SDK）（选接）	22
4.11	显示悬浮窗（必接）	22
4.12	关闭悬浮窗（必接）	23
4.13	获取公告（必接）	23
4.14	退出游戏（必接）	24
4.15	onResume（必接）	24
4.16	onPause（必接）	25
4.17	查询登录用户实名认证状态（选接）	25
4.18	自定义事件统计（选接）	26
5、	录屏（选接）	27
6、	混淆	30

1、接入准备

1.1 环境

- 当您下载完 BDGameSDK.rar 包后，解压出的 BDGameSDK_LibProject 即为百度移动游戏 SDK 开发包。工程目录如下：



- BDGameSDK_LibProject 工程里面的资源（即 res 文件夹下所有文件包括 drawable,id,string,color,layout 等）分别以：“dk_”，“ebpay_”，“tieba_”，“bd_”，“bdp_”开头命名。CP 接入时请不要以这些为开头命名资源，以避免冲突。

- 开发包支持在 Android SDK2.3 及以上固件版本运行，编译时请用 Android SDK4.0 以上版本编译。

1.2 参数

在接入 SDK 前请先向开发者后台申请：

- AppId: 游戏 AppId，SDK 初始化接口用到，详情查看[初始化](#)。
- AppKey: 游戏 AppKey，SDK 初始化接口用到，详情查看[初始化](#)。
- SecretKey: CP 游戏与服务端通信校验用（由于涉及到加密通信, CP 必须严格对 SecretKey 进行保密，仅服务端使用，不能写到客户端代码中）。

1.3 附加信息

SDK 需要用到的 APPID、APPKEY、APPSECRET 请在百度移动游戏开发者平台新注册或使用已有的百度账号登录后点击“管理中心”进入[联运游戏]添加游戏后获得。一款游戏只允许使用一个 APPID，游戏上架后 APPID 正常情况下不允许改变。

开发者平台地址：<http://app.baidu.com>

您在接入中遇到问题可访问：<http://bbs.mgame.baidu.com>论坛

2、SDK 配置

2.1 Application 配置

您需要在您的游戏里定义一个类来继承 Application，然后在该类里初始化游戏 SDK 的 **initApplication(Application)**接口，并将该类配置到 AndroidManifest.xml 里面，举例如下：

```
public class DemoApplication extends Application{
    @Override
    public void onCreate() {
        super.onCreate();
        com.baidu.gamesdk.BDGameSDK.initApplication(this);
    }
}
```

```
<application
    android:name="com.baidu.bdgamesdk.demo.application.DemoApplic
ation">
```

2.2 SDK 接入类型配置

SDK 接入类型有三种：百度，多酷，91，默认是百度。

若是新游戏：使用百度类型，或者不配置，此时返回百度用户 ID。

若兼容多酷老游戏（指的是之前已经接入多酷游戏 SDK）：使用多酷类型，此时返回多酷用户 ID。

若兼容 91 老游戏（指的是之前已经接入 91 游戏 SDK）：使用 91 类型，此时返回 91 用户 ID。

AndroidManifest.xml 配置如下：

```
<!-- 0-百度 1-多酷 2-91 -->
<meta-data android:name="BDPlatformType" android:value="0"/>
```

2.3 Permission、Activity、Service 等相关配置

请将 BDGameSDK_LibProject/AndroidManifest.xml 的相关 permission, activity, service, receiver 等拷贝到您游戏 AndroidManifest.xml。

1、请将 AndroidManifest.xml 中所有的 package 都改为您游戏的包名。

例如：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.baidu.gamesdk.demo" //对应您游戏的包名
    android:versionCode="1"
    android:versionName="1.0" >
```

2、AndroidManifest.xml 中包含前缀的修改，请将所有 packagename 都改为您游戏的包名若您游戏的包名为：com.baidu.bdgamesdk.demo，例如则修改前：

```
<!-- ↓↓↓ QQ插件 需修改<data android:scheme="qwallet"+packagename />
替换packagename为应用包名，如： <data android:scheme="qwalletcom.baidu.test"
否则将无法收到支付结果通知！！ />↓↓↓ -->
<activity
    android:name="com.baidu.platformsdk.pay.channel.qqwallet.QQPayActivity"
    android:launchMode="singleTop"
    android:exported="true"
    android:configChanges="orientation/navigation/screenSize/keyboard/keyboardHidden"
    android:theme="@android:style/Theme.Translucent.NoTitleBar">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.BROWSABLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="qwallet+packagename" />
    </intent-filter>
</activity>

<!-- ↑↑↑ QQ 插件 ↑↑↑ -->
```

修改后为：

```
<!-- ↓↓↓ QQ插件 需修改<data android:scheme="qwallet"+packagename />
替换packagename为应用包名，如： <data android:scheme="qwalletcom.baidu.test"
否则将无法收到支付结果通知！！ />↓↓↓ -->
<activity
    android:name="com.baidu.platformsdk.pay.channel.qqwallet.QQPayActivity"
    android:launchMode="singleTop"
    android:exported="true"
    android:configChanges="orientation/navigation/screenSize/keyboard/keyboardHidden"
    android:theme="@android:style/Theme.Translucent.NoTitleBar">
```



```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.BROWSABLE" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="qwalletcom.baidu.bdgamesdk.demo" />
</intent-filter>
</activity>

<!-- ↑↑↑ QQ 插件 ↑↑↑ -->
```

3、请修改 AndroidManifest.xml 中 provider android:authorities 属性。

例如：修改前

```
<!-- 需修改android:authorities="packagename" 替换packagename为应用包名，如：
android:authorities="com.baidu.test" -->
<provider
    android:name="com.duoku.platform.download.DownloadProvider"
    android:authorities="packagename"
    android:exported="false" />
```

修改后

```
<!-- 需修改android:authorities="packagename" 替换packagename为应用包名，如：
android:authorities="com.baidu.test" -->
<provider
    android:name="com.duoku.platform.download.DownloadProvider"
    android:authorities="com.baidu.bdgamesdk.demo"
    android:exported="false" />
```

AroidManifest.xml 文件的所有修改在注释中已经详细标识出来，请对应修改。

4、添加权限

```
<!-- ↓↓↓ SDK ↓↓↓ -->
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.GET_TASKS" />

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

```
<!-- 百度钱包 -->
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<!-- 百度钱包 -->
<uses-permission android:name="android.permission.READ_SMS" />
<!-- 百度钱包 -->

<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />

<!-- 若需录音需加此权限 omlet sdk -->
<!-- <uses-permission android:name="android.permission.RECORD_AUDIO"/> -->
<!-- ↑↑↑ SDK ↑↑↑ -->

<!-- ↓↓↓ DK SDK ↓↓↓ -->
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_DOWNLOAD_MANAGER"
/>
<uses-permission
android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />

<!-- ↑↑↑ DK SDK ↑↑↑ -->
```

2.4 assets 配置

请将 BDGameSDK_LibProject/assets 文件夹下的文件拷贝到您游戏工程对应的 assets 文件夹里。

2.5 debug 模式

可通过 SDK [初始化](#) 接口的 `BDGameSDKSetting` 参数，设置 SDK 的 `Domain.DEBUG` 模式。在该模式下您可：

1. 可通过[支付](#)接口传入 `debugCallbackUrl` 来[模拟测试](#)支付结果通知(SDK 服务端会在支付成功时链接该 `Url`)，正式的支付结果通知地址需要在开发者后台配置。在 `Domain.DEBUG` 模式下，如果 `debugCallbackUrl` 为空，则支付结果会根据开发者后台配置的地址通知；而在 `Domain.RELEASE` 模式下，不管 `debugCallbackUrl` 有没有值，支付结果都会根据开发者后台配置的地址通知。

2. 在您的 APP 发布前切记在 SDK 的[初始化接口](#)，将 `Domain.DEBUG` 设置去掉（SDK 默认为 `Domain.RELEASE` 模式），或者改为设置为 `Domain.RELEASE`。

3、API 接入说明

3.1 API 接入顺序

- 在调用其他 API 前需先调用[初始化接口](#)对 SDK 进行初始化。
- 在完成[初始化接口](#)调用后需调用[公告接口](#)，此接口需在调用[登录接口](#)前调。
- 对于有依赖于登录用户的接口如支付，获取用户 `UID` 等需要先调用[登录接口](#)登录，您需要在[登录接口](#)的回调及[设置切换账号结果通知](#)回调里面实现游戏的登录相关逻辑，具体请查看对应接口的功能介绍。
- 您可以[调用设置会话失效监听接口](#)，用来在登录会话失效时捕获到失效通知，以便保存玩家的游戏数据或者重新调用登录接口走登录逻辑。
- 在游戏退出时需要先调用[退出游戏接口](#)，然后在该接口的通知回调里处理游戏退出逻辑。

3.2 必须接入的 API 接口

您须在您的游戏里面加入：

- 在启动游戏时调用[初始化](#)
- 初始化成功之后用户登录之前，在游戏主 Activity 调用[获取公告](#)
- 在初始化成功后调用[登录](#)
- 在登录成功后获取[用户 UID](#)和[用户 AccessToken](#)
- 登录成功调用[支付](#)
- 在游戏主 Activity 设置[会话失效](#)
- 在游戏主 Activity 设置[切换账号](#)
- 登录成功可调用[退出游戏](#)
- 在游戏主 Activity 的 onPause 方法中调用 [onPause 接口](#)、在游戏 onResume 方法中调用 [onResume 接口](#)。
- 登录成功后，在游戏主界面[显示悬浮窗](#)，退出时[关闭悬浮窗](#)。

注意：必须接入的接口实现见 **demo**，实际接入过程中如果必接接口中有少接和漏接则会导致程序功能缺失、逻辑异常等情况。

3.3 API 回调接口说明

在 SDK 提供的 API 中，有一部份 API 带有回调接口，如：[初始化](#)，[登录](#)，[支付](#)等。该回调接口为 `IResponse<T>`。CP 需要实现其中的 `onResponse` 方法：

```
void onResponse(int resultCode, String resultDesc, T extraData)
```

- **resultCode**: 接口状态码；CP 可以根据该状态码实现自己业务逻辑，接口状态码在下面 API 功能介绍时的举例内容里面都有定义，详细请查看各个 API 的功能介绍。
- **resultDesc**: 接口状态对应的描述；该字段纯文本描述，为 **resultCode** 的补充说明。
- **extraData**: API 返回的数据对象；该字段为泛型，具体类型在各个 API 中定义。该字段值在接口请求成功时返回。当 **extraData** 被定义为 **Void** 时，无需理会该字段；详细请查看各个 API 的功能介绍。

4、API 功能介绍

4.1 初始化（必接）

接口：

```
static void init(Activity activity, BDGameSDKSetting setting,
IResponse<Void> responser)
```

IResponse 回调接口状态码(resultCode)为：

状态码	描述
ResultCode. <i>INIT_SUCCESS</i>	初始化成功
ResultCode. <i>INIT_FAIL</i>	初始化失败

调用举例：

```
BDGameSDKSetting mBDGameSDKSetting = new BDGameSDKSetting();
mBDGameSDKSetting.setAppID(3067515); //APPID设置
mBDGameSDKSetting.setAppKey("f30s4GA0qxgm79GqbnkT9L8T"); //APPKEY设置
mBDGameSDKSetting.setDomain(Domain.RELEASE); //设置为正式模式
mBDGameSDKSetting.setOrientation(Orientation.LANDSCAPE); //设置为横屏
BDGameSDK.init(this, mBDGameSDKSetting, new IResponse<Void>(){
```

```

@Override
public void onResponse(int resultCode, String resultDesc,
                      Void extraData) {
    switch(resultCode){
        case ResultCode.INIT_SUCCESS:
            //初始化成功
            Intent intent = new Intent(WelcomeActivity.this,
MainActivity.class);
            startActivity(intent);
            finish();
            break;

        case ResultCode.INIT_FAIL:
        default:
            Toast.makeText(WelcomeActivity.this, "启动失败",
Toast.LENGTH_LONG).show();
            finish();
            //初始化失败
        }
    }
});

```

4.2 登录（必接）

接口：

```
static void login(IResponse<Void> responser)
```

IResponse 回调接口状态码(resultCode)为：

状态码	描述
ResultCode.LOGIN_SUCCESS	登录成功
ResultCode.LOGIN_CANCEL	取消登录
ResultCode.LOGIN_FAIL	登录失败

调用举例：

```
BDGameSDK.Login(new IResponse<Void>() {  
  
    @Override  
    public void onResponse(int resultCode, String resultDesc,  
Void extraData) {  
        String hint = "";  
        switch(resultCode){  
            case ResultCode.LOGIN_SUCCESS:  
                hint = "登录成功";  
  
                break;  
            case ResultCode.LOGIN_CANCEL:  
                hint = "取消登录";  
                break;  
            case ResultCode.LOGIN_FAIL:  
            default:    //其他值为登录失败  
                hint = "登录失败";  
        }  
    }  
});
```

4.3 是否已登录（选接）

接口：

```
Static boolean isLogined()
```

用于获取用户是否处于登录状态。

4.4 登录用户 UID（必接）

接口：

```
static String getLoginUid()
```

依赖[登录](#)接口,在登录回调成功后可以调用此接口获取登录用的 uid。

4.5 登录用户 AccessToken（必接）

接口：

```
static String getLoginAccessToken()
```

依赖[登录](#)接口，在登录回调成功后可以调用此接口获取登录用的 AccessToken。

AccessToken 可用于百度游戏 SDK 服务器支付结果通知 CP 应用服务器的数据校验。具体可查看《[百度游戏 SDK 服务端接入协议.pdf](#)》文档。

4.6 注销（选接）

接口：

```
static void logout()
```

调用注销接口 `logout` 后当前用户登录状态失效。cp 可以清除用户游戏登录状态，再次获取用户信息进行操作需要调用[登录接口](#)。

4.7 支付（必接）

接口：

```
static boolean pay(PayOrderInfo payOrderInfo,String  
debugCallbackUrl,IResponse<PayOrderInfo>responser)
```

依赖登录接口。

● PayOrderInfo: 支付订单信息。有以下属性：

属性	类型	描述
cooperatorOrderSerial	String	CP 订单号，具有唯一性，不超过 256 个字符（限：字母、数字、_、-）
productName	String	商品名称，不超过 50 个字符（限：中文、字母、数字、_、-）

totalPriceCent	long	支付总金额，以“分”为单位
ratio	int	商品兑换比例，如：兑换比例为 5 时，表示 100 分钱可兑换 5 个商品； （该字段只在非定额支付时生效，定额支付不生效）
extInfo	String	扩展字段；该字段将会在支付成功后原样返回给 CP(不超过 500 个字符)
uid	String	登录回调成功后获取的用户 id(详见 sdk demo)。

- debugCallbackUrl: CP 设置 debug 模式下，可通过此参数设置回调地址，来模拟支付接口通知，届时，服务端会在支付成功后会调用该地址。详情查看 [debug 模式](#)。
- IResponse<PayOrderInfo>: 支付结果通知回调接口。

IResponse 回调接口状态码(resultCode)为:

状态码	描述
ResultCode.PAY_SUCCESS	支付成功
ResultCode.PAY_CANCEL	取消支付
ResultCode.PAY_FAIL	支付失败
ResultCode.PAY_SUBMIT_ORDER	订单已经提交，支付结果未知, 百度游戏 SDK 服务端在订单支付成功后会通知 CP 的游戏应用服务端，服务端对接请查看：《百度 SDK 服务端接入协议.pdf》文档。

支付分为定额支付和非定额支付两种方式:

4.7.1 定额支付

指 CP 传入的支付金额大于 0(即 PayOrderInfo 里面的 totalPriceCent

大于 0) 时，即为定额支付。此时订单信息里的商品兑换比例无用。

商品名称：100金币

应付金额：1元

调用举例：

生成订单：

```
PayOrderInfo payOrderInfo = new PayOrderInfo();
String cpOrderId = UUID.randomUUID().toString();
payOrderInfo.setCooperatorOrderSerial(cpOrderId); //CP订单号
payOrderInfo.setProductName("100金币"); //商品名称
payOrderInfo.setTotalPriceCent(100); //以分为单位，long类型
payOrderInfo.setRatio(1); //兑换比例，此时不生效
payOrderInfo.setExtInfo(""); //该字段在支付通知中原样返回，不超过500个字符
```

```
BDGameSDK.pay(payOrderInfo, null,
    new IResponse<PayOrderInfo>(){

    @Override
    public void onResponse(int resultCode, String
resultDesc, PayOrderInfo extraData) {
        alertDialog.cancel();
        String resultStr = "";
        switch(resultCode){
            case ResultCode.PAY_SUCCESS: //支付成功
                resultStr = "支付成功:" + resultDesc;
                break;
            case ResultCode.PAY_CANCEL: //订单支付取消
                resultStr = "取消支付";
                break;
            case ResultCode.PAY_FAIL: //订单支付失败
                resultStr = "支付失败: " + resultDesc;
                break;
            case ResultCode.PAY_SUBMIT_ORDER: //订单已经提交，支付结果未知
                // (比如：已经请求了，但是查询超时)
```

```
        resultStr = "订单已经提交，支付结果未知";  
        break;  
    }  
}  
});
```

4.7.2 非定额支付

指 CP 传入的支付金额等于 0（即 PayOrderInfo 里面的 totalPriceCent 等于 0）时，即为非定额支付。此时 CP 需要在商品订单信息里设置商品兑换比例，否则订单无效。

商品兑换比例表示：每 100 分（即 1 元）可兑换 x 个商品。如兑换比例为 10 时，表示 100 分钱可兑换 10 个商品。

请选择充值金额：(1元=10金币)

10元	30元	50元	100元
200元	500元	1000元	2000元

其他金额(元)

调用举例：

```
//此时1元RMB等于10个金币  
PayOrderInfo payOrderInfo = new PayOrderInfo();  
String cpOrderId = UUID.randomUUID().toString();  
payOrderInfo.setCooperatorOrderSerial(cpOrderId); //CP订单号  
payOrderInfo.setProductName("金币"); //商品名称
```

```
payOrderInfo.setTotalPriceCent(0); //非定额支付这里填0
payOrderInfo.setRatio(10); //兑换比例
payOrderInfo.setExtInfo(""); //该字段在支付通知中原样返回,不超过500个字符
```

4.8 设置切换账号结果通知（必接）

接口：

```
static void setSuspendWindowChangeAccountListener(
    IResponse<Void> changeAccountResponder)
```

IResponse 回调接口状态码(resultCode)请参考[登录接口](#)。

调用举例：

```
BDGameSDK.setSuspendWindowChangeAccountListener(new IResponse<Void>(){

    @Override
    public void onResponse(int resultCode, String resultDesc,
        Void extraData) {
        switch(resultCode){
            case ResultCode.LOGIN_SUCCESS:
                //TODO 登录成功，不管之前是什么登录状态，游戏内部都要切换成新
                的用户
                break;
            case ResultCode.LOGIN_FAIL:
                //TODO 登录失败，游戏内部之前如果是已经登录的，要清楚自己记录的登
                录状态，设置成未登录。如果之前未登录，不用处理。
                break;
            case ResultCode.LOGIN_CANCEL:
                //TODO 操作前后的登录状态没变化
                break;
        }
    }

});
```

4.9 设置会话失效监听（必接）

接口：

```
static void setSessionInvalidListener(IResponse<Void> response)
```

IResponse 回调接口状态码(resultCode)为：

状态码	描述
ResultCode. SESSION_INVALID	会话失效，CP 在捕获到该状态码后可走重新登录逻辑。

会话失效的场景：

1. 该用户在其他设备，同一款游戏登陆。
2. 该用户同SDK服务器会话链接长久失联。
3. 该用户被SDK服务器踢下线。
4. 该用户修改密码

触发该接口回调的场景：

在会话失效的情况下，调用SDK API请求SDK服务器或者SDK自动触发请求SDK服务器。

调用举例：

```
BDGameSDK.setSessionInvalidListener(new IResponse<Void>(){

    @Override
    public void onResponse(int resultCode, String resultDesc,
        Void extraData) {
        if(resultCode == ResultCode.SESSION\_INVALID){
            Toast.makeText(getApplicationContext(), "会话失效，
请重新登录", Toast.LENGTH\_LONG).show();
            //TODO 此处CP可调用登录接口
        }
    }

});
```

4.10 设置多酷 AppId 和 AppKey(兼容多酷手游 SDK)(选接)

接口：

```
static void oldDKSdkSetting(String dkAppId, String dkAppKey)
```

dkAppId 和 dkAppKey 为接入多酷手游 SDK 时的 AppId 和 AppKey。

(该接口仅在需要兼容多酷游戏 SDK 时调用)

APP ID: 1111111 (DKAPP ID:1234)

APP KEY: XXXXXXXXXXXXXXXX (DKAPP KEY: YYYYYYYYYYYYYYYY)

- 原先使用多酷手游SDK的游戏，需在此写入“多酷AppId和AppKey”，否则更换为百度游戏SDK后不能使用论坛、专区等功能。
- 判断标准：如果“多酷AppId和AppKey”在开发者平台显示（如上图所示），则需要设置此接口；如果没有显示，则无需设置该接口。
- 该接口的调用必须放在[初始化接口](#)前，否则调用无效。

4.11 显示悬浮窗（必接）

接口：

```
static void showFloatView(Activity activity)
```

- Activity: 悬浮窗显示在游戏主 Activity 上。

注意：在整个游戏接入过程中显示悬浮窗只在游戏主 Activity 调用一次。详见 demo 中 [GameActivity](#)

- 本接口依赖于[登录](#)。需要在登录成功之后调用否则无法成功调用。

4.12 关闭悬浮窗（必接）

接口：

```
static void closeFloatView(Activity activity)
```

- 关闭悬浮窗，

注意：在整个游戏接入过程中关闭悬浮窗只调用一次，详见 [demo](#)：

```
@Override
public void finish() {
    // TODO Auto-generated method stub
    super.finish();
    BDGameSDK.closeFloatView(this); // 关闭悬浮窗
}
```

4.13 获取公告（必接）

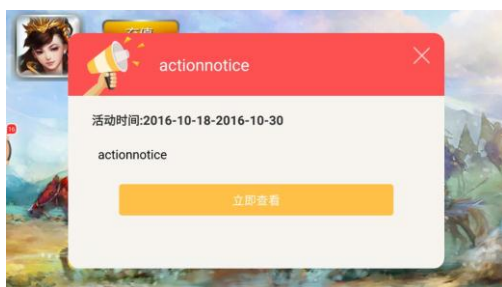
接口：

```
static void getAnnouncementInfo(Activity activity)
```

获取公告在[初始化](#)成功后，调用[登录接口](#)之前由 cp 主动调用，具体可参考 demo 中的 MainActivity 如下：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    initView();
    BDGameSDK.getAnnouncementInfo(this);
}
```

调用[获取公告](#)接口成功，在登录完成后会显示出公告页面，如下图：



注意：公告功能需要在后台配置后才能生效，具体测试方案需要在提包前联系相关接入同学进行配置测试。

4.14 退出游戏（必接）

接口：

```
static void gameExit(Context context, OnGameExitListener onGameExitListener)
```

在您游戏退出前请先调用此 API，然后在 `OnGameExitListener` 的回调 `onGameExit()` 方法里再处理您的游戏退出逻辑。

调用举例：

```
BDGameSDK.gameExit(this, new OnGameExitListener(){
    @Override
    public void onGameExit() {
        // TODO 在此处执行您的游戏退出逻辑
        finish();
    }
});
```

4.15 onResume（必接）

接口：

```
static void onResume(Activity activity)
```


在您 Activity 的 onResume 里调用，具体详细见 demo 中的 BaseActivity，如下：

```
@Override
protected void onResume() {
    super.onResume();
    BDGameSDK.onResume(this);
}
```

4.16 onPause（必接）

接口：

```
static void onPause(Activity activity)
```

在您 Activity 的 onPause 里调用，具体详细见 demo 中的 BaseActivity，，如下：

```
@Override
protected void onPause() {
    super.onPause();
    BDGameSDK.onPause(this);
}
```

4.17 查询登录用户实名认证状态（选接）

接口：

```
static boolean queryLoginUserAuthenticateState(Context context,
        IResponse<Integer> callback)
```

- IResponse<Integer>:通过该回调接口返回的状态为：

0	该用户未进行实名认证
1	该用户已认证：未成年
2	该用户已认证：成年

接口调用举例：

```
BDGameSDK.queryLoginUserAuthenticateState(context, new

IResponse<Integer>(){

    @Override
    public void onResponse(int resultCode, String resultDesc,
        Integer extraData) {
        if(resultCode == ResultCode.RESULT_OK){// 查询成功
            switch(extraData.intValue()){
                case 0:
                    //该用户未进行实名认证
                    break;
                case 1:
                    //该用户已认证：未成年
                    break;
                case 2:
                    //该用户已认证：成年
                    break;
            }
        } else {
            // 查询失败
        }

    }

});
```

4.18 自定义事件统计（选接）

接口：

```
static void onTag(String eventId, Map<String, String> map)
```

eventId: 事件类型标识

map: 事件的属性名称和值

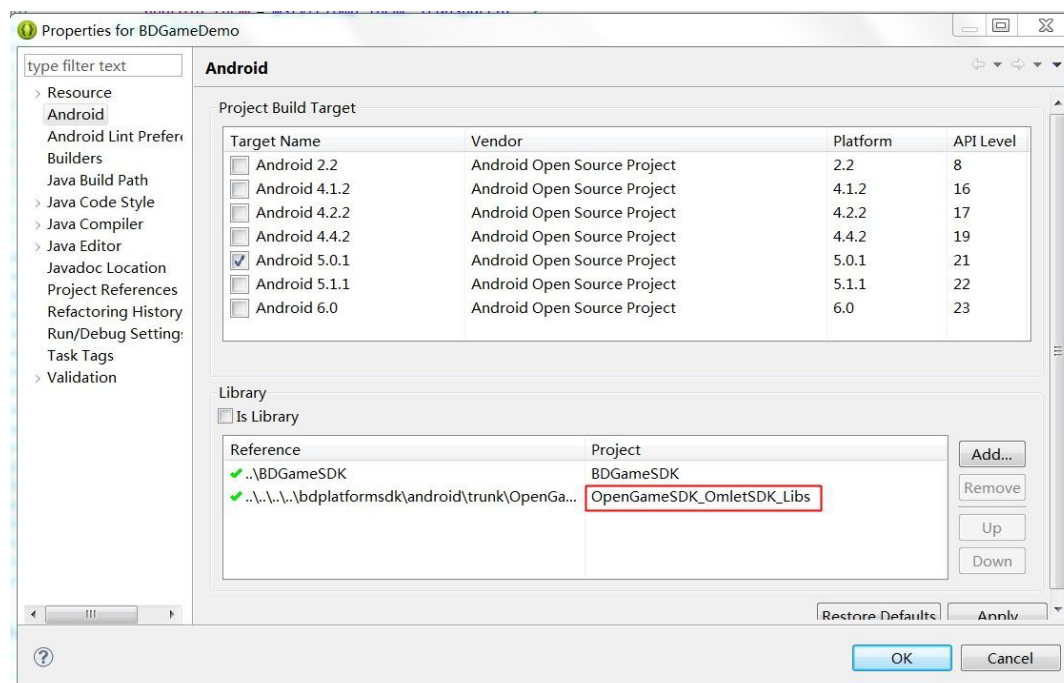
接口调用举例:

```
private void onTag() {  
    Map<String, String> map = new HashMap<String,  
String>();  
    map.put("custom key 1", "custom value 1");  
    map.put("custom key 2", "custom value 2");  
    map.put("custom key 3", "custom value 3");  
    BDGameSDK.onTag("eventId", map);  
}
```

5、录屏（选接）

CP 可以根据实际情况选择是否接入录屏功能，录屏功能可以在玩家游戏过程中进行录制，接入方式如下：

1、在游戏工程中引用录屏工程，如下图：



2、修改AndroidManifest文件内容，支持sdk版本`android:minSdkVersion="18"`

详见 demo，如下图：

```
<!-- ↓↓↓ 录屏 ↓↓↓ -->
    <activity
        android:name="mobisocial.omlet.overlaybar.ui.activity.SigninActivity"
        android:hardwareAccelerated="true"
        android:screenOrientation="sensorLandscape"
        android:theme="@style/Omp.Theme.Transparent" >
    </activity>
    <activity
android:name="mobisocial.omlet.overlaybar.ui.activity.UploadCompleteActivity"
        android:hardwareAccelerated="true"
        android:screenOrientation="sensorLandscape"
        android:theme="@style/Omp.Theme.Dialog" >
    </activity>
    <activity
android:name="mobisocial.omlet.overlaybar.ui.activity.VideoEditorActivity"
        android:configChanges="orientation|screenSize"
        android:hardwareAccelerated="true"
        android:screenOrientation="sensorLandscape"
        android:theme="@style/Omp.Theme.Transparent" >
    </activity>
    <activity
android:name="mobisocial.omlet.overlaybar.ui.activity.ScreenshotEditActivity"
        android:hardwareAccelerated="true"
        android:screenOrientation="sensorLandscape"
        android:theme="@style/Omp.Theme.Transparent" >
    </activity>
    <activity
android:name="mobisocial.omlet.overlaybar.ui.activity.MediaUploadActivity"
        android:configChanges="orientation|screenSize"
        android:exported="false"
        android:hardwareAccelerated="true"
        android:screenOrientation="sensorLandscape"
        android:theme="@style/Omp.Theme.Transparent" >
    </activity>
    <service
android:name="mobisocial.omlet.overlaybar.util.FileUploadIntentService"
        android:exported="false" >
    </service>
```

```
<service
    android:name="mobisocial.omlib.service.OmLibService"
    android:exported="false"
    android:label="Omlet Network Service" />

<!--需修改android:authorities="packagename.provider"
替换packagename为应用包名，如：
android:authorities="com.baidu.test.provider"-->
<provider
    android:name="mobisocial.omlib.service.OmLibContentProvider"
    android:authorities="com.baidu.bdgamesdk.demo.provider" />

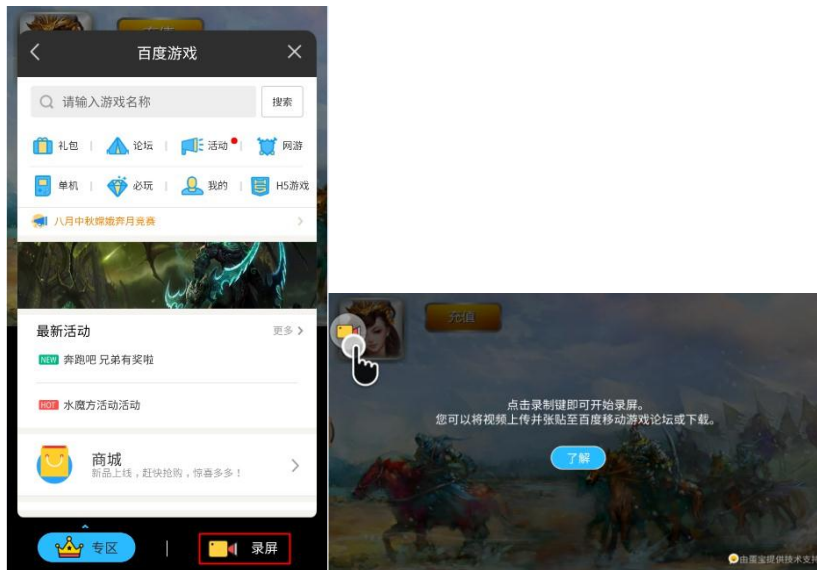
<!--需修改android:authorities="packagename.initializer"
替换packagename为应用包名，如：
android:authorities="com.baidu.test.initializer"-->
<provider
    android:name="glrecorder.Initializer"
    android:authorities="com.baidu.bdgamesdk.demo.initializer"
    android:exported="false"
    android:initOrder="2147483647" />

<!-- ↑↑↑ 录屏 ↑↑↑ -->
```

3、添加权限如下：

```
<!-- 若需录音需加此权限 omlet sdk -->
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<!-- ↑↑↑ SDK ↑↑↑ -->
```

4、运行游戏后进入悬浮窗开始录制,如下：



6、混淆

百度移动游戏 SDK 开发包是以 jar 包及资源文件提供给 CP 的，其中 jar 包已是半混淆状态，您在混淆自己 APK 包的时候请不要将百度移动游戏 SDK 的 jar 包一起混淆，因为里面有些自定义 UI 控件，若被混淆后会因为无法找到相关类而抛异常。

您可以在用 ant 构建混淆包的 build.xml 里面对应位置或者在 proguard.cfg 里加入：

```
-dontwarn com.baidu.**
-dontwarn com.alipay.**
-dontwarn com.duoku.**
-dontwarn android.content.pm.**
-dontwarn com.slidingmenu.**
-dontwarn com.unionpay.**
-dontwarn com.UCMobile.**

-keep public class com.baidu.** {
    * ;
}
```

```
-keep public class com.alipay.** {  
    * ;  
}  
-keep public class com.duoku.**{  
    * ;  
}  
-keep public class android.content.pm.**{  
    * ;  
}  
-keep public class com.slidingmenu.** {  
    * ;  
}  
-keep public class com.unionpay.** {  
    * ;  
}  
-keep public class com.UCMobile.**{  
    * ;  
}
```

以避免百度移动游戏 SDK 的 jar 包被混淆。