

优酷游戏 SDK 接入文档

Sdk 版本号：2.2

优酷游戏

修改记录

修订人	版本	备注	修订时间
王洪波	0.1	初稿	2014-03-18
徐琦	0.2	初稿	2014-03-21
王洪波	0.3	demo 流程变化。对应 demo 版本号 1	2014-03-26
王宝振	0.4	Cp 获取 apporderID，支付成功通知 Cp	2014-03-26
王洪波	0.5	提供游戏后台通过 session 验证用户信息	2014-04-01
王宝振	0.6	删除 cp 获取 apporderID 接口	2014-04-01
王洪波	0.7	autologin 接口，只有用户 back 键返回取消登录时，才会回调失败接口	2014-04-02
赵元青	0.8	增加通过 sessionid 验证用户有效性接口	2014-04-02
赵元青	0.9	签名算法笔误	2014-04-03
徐琦	1.0	添加 WapPayActivity 的声明、注明消费以“分”为单位	2014-04-09
王宝振	1.1	修改通知 CP 返回报文为标准 JSON	2014-04-09
徐琦	1.2	整理目录顺序，将登录流程整合到一起，修正图片角标	2014-04-11
王洪波	1.3	代码加边框底纹修饰。只能在 ui 线程中调用的接口加提示	2014-04-14
王洪波	1.4	logout 接口加标注，当且仅当在切换账号的时候调用。	2014-04-15
徐琦	1.5	完善支付参数的注释	2014-04-24
王宝振	1.6	增加 apporderID 长度限制	2014-04-28
王洪波	1.7	新增 SDK 测试环境切换接口	2014-04-29
徐琦	1.8	新增短代支付,需要申请 manifest 权限	2014-06-12
王宝振	1.9	支付成功通知接口增加短代支付独有的两个参数	2014-06-12
徐琦	2.0	加入极简收银台	2014-07-10
赵元青	2.1	Hmac 算法备注	2014-07-30
王宝振	2.2	参数备注	2014-08-04
徐琦	2.3	1.8 增加透传参数说明,manifest 中的注册文档修改	2014-09-02
张朋	2.4	包名后缀要求	2014-09-03
王宝振	2.5	支付通知增加透传参数	2014-09-17
王洪波	2.6	加入单机版游戏接入方式	2014-09-17

优酷游戏 SDK 接入文档

王洪波	2.7	补充切换账号描述	2014-10-22
徐琦	2.8	1.9.0 版本的改动 (增加悬浮窗接口)	2014-10-29
王洪波	2.9	1.9.0 版本 (增加退出游戏接口)	2014-10-29
王洪波	3.0	游戏版本号长度限定	2014-11-27
徐琦	3.1	1.9.2 版本增加了悬浮窗切换账号的回调功能, 自定义控件 editview 改成可以复制出来的控件	2014-12-03
王洪波	3.2	Privatekey 统一, 后台申请 privatekey	2014-12-11
徐琦	3.3	悬浮窗接口改变, PAYKEY 为后台加密密钥	2015-01-15
徐琦	3.4	单机游戏的设置接口应为 YKPlatform.setStandAlone(true);	2015-01-26
于桐	3.5	增加崩溃统计接口	2015-06-08
于桐	3.6	增加会员特权页面接口	2015-07-27
杨时茂	3.6	增加开通会员成功通知 cp 接口	2015-07-27
徐琦	3.7	增加微信支付	2015-08-07
于桐	3.8	增加推送消息相关的权限和服务等的声明	2015-08-27

1. 目录

1. 目录.....	4
1. 概述.....	5
1.1 注意事项.....	5
2. 接入准备.....	5
2.1 接入之前.....	5
2.1.1 工程依赖.....	5
2.1.2 微信支付接入.....	7
2.1.3 配置 AndroidManifest.....	7
2.2 登录接口及流程.....	12
2.2.1 调用 SDK 登录接口,获取 sessionid.....	12
2.2.2 通过 SessionID 验证用户有效性接口.....	14
2.3 优酷游戏 SDK 提供给游戏开发者的其他接口.....	14
2.3.1 设置单机游戏.....	14
2.3.2 初始化.....	15
2.3.3 注销（切换账号）.....	15
2.3.4 退出游戏.....	16
2.3.5 取 sdk 版本号.....	16
2.3.6 支付.....	16
2.3.7 支付成功通知 CP 接口.....	17
2.4 悬浮窗接口.....	18
2.4.1 优酷 sdk 悬浮窗开关接口.....	18
2.5 崩溃统计接口.....	19
2.5.1 由 sdk 内部进行崩溃统计.....	19
2.5.2 由 cp 接入时调用 sdk 提供的接口进行统计.....	20
2.6 会员特权页面开启接口.....	21
2.6.1 由 cp 接入会员特权入口.....	21
2.7 开通会员成功通知 CP 接口.....	22

1.概述

1.1 注意事项

- A. Packagename 包名加后缀.youku
- B. 代码混淆时不要混编 jar 包中的类
- C. 在需要调用 SDK 接口的 Activity，需要在 ActivityManifest 中配置属性
android:configChanges="fontScale|orientation|keyboardHidden|locale|navigation|screenSize|uiMode"
- D. 单机游戏请忽略 2.2 章。
- E. 游戏版本号 versionCode 限制长度不能大于 4 位。

2.接入准备

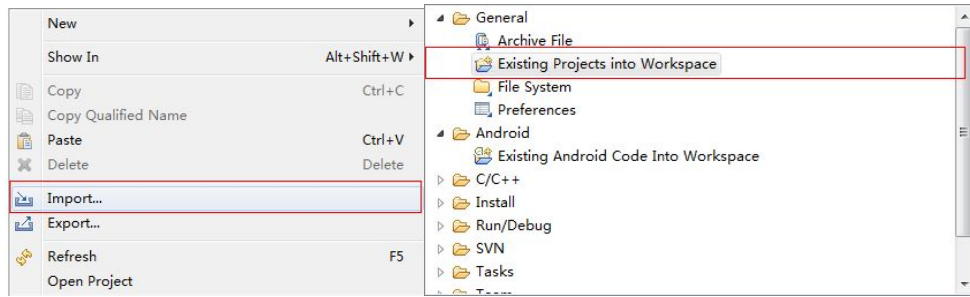
2.1 接入之前

开发者在调用接口之前，需要与优酷游戏商务沟通申请 APPID、APPKEY、PRIVATEKEY 和 PAYKEY。其中 APPID、APPKEY 是应用的唯一标识，PRIVATEKEY 相当于是应用的密钥，PAYKEY 是保存在游戏后台的密钥，用于游戏后台跟优酷后台交互。APPKEY 与应用是一对一的关系，一个 APPKEY 只能分配给一个应用使用。若多个应用使用一个 APPKEY 会导致不能正常计费，影响收入；还会导致 SDK 升级失败、各项统计数据出错等严重问题，影响开发者的收入和用户体验。并且 APPID 与 APPKEY 必须对应统一。

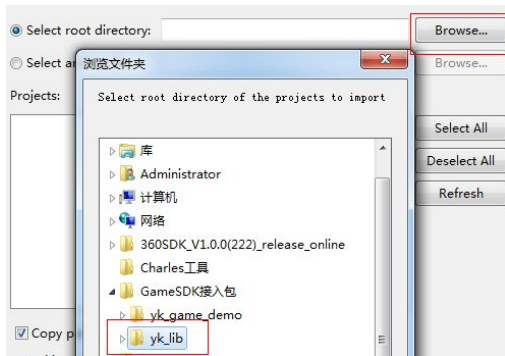
为了安全，PAYKEY 不能写进手机应用里，只能存储在应用服务器上，以防反编译手机应用获取此信息，PAYKEY 泄漏可能用于伪造支付成功通知消息，导致合作双方经济受损。PAYKEY 不允许保存在客户端，不能泄露。

2.1.1 工程依赖

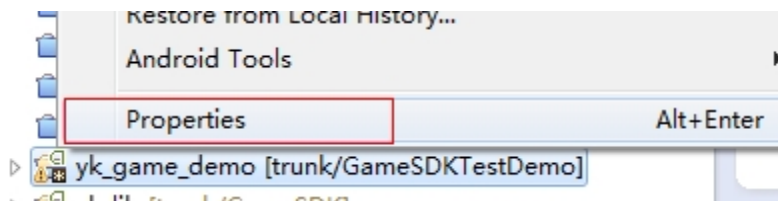
- 1.将 yk_lib 工程导入 eclipse 工程中，如（图 2.1.3_1）、（图 2.1.3_2）、（图 2.1.3_3）；
- 2.将 yk_lib 添加为游戏的依赖工程，如（图 2.1.3_4——2.1.3_7）；



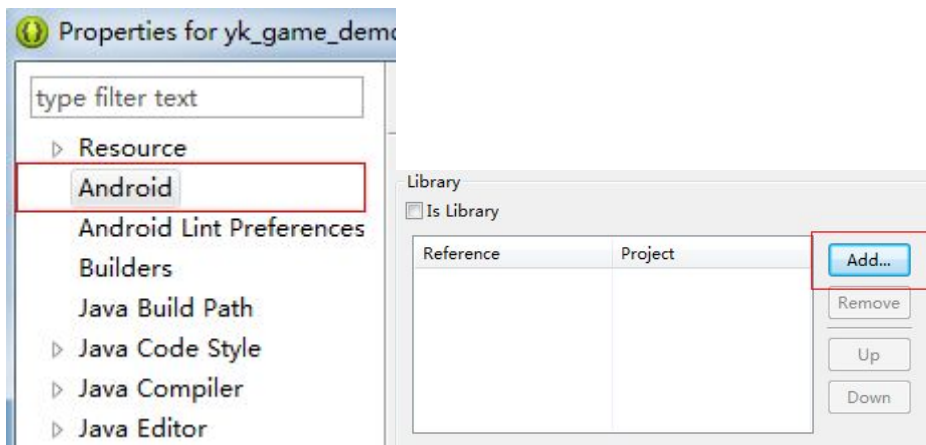
(图 2.1.3_1) (图 2.1.3_2)



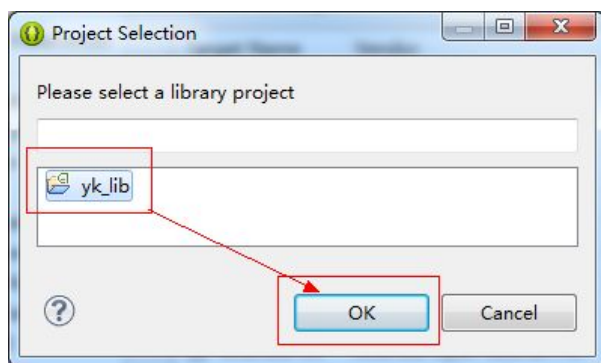
(图 2.1.3_3)



(图 2.1.3_4)



(图 2.1.3_5) (图 2.1.3_6)



(图 2.1.3_7)

2.1.2 微信支付接入

*不接入微信的游戏方可忽略此条（是否接入微信需询问商务人员）

在游戏工程包名目录下添加子包名 `wxapi`，将优酷游戏 demo 工程下的 `WXPayEntryActivity.java` 复制到游戏目录下。`WXPayEntryActivity.java` 为优酷微信支付的回调类。

2.1.3 配置 AndroidManifest

A 添加权限

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.READ_SETTINGS" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.READ_LOGS" />
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.BROADCAST_STICKY"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="com.android.launcher.permission.READ_SETTINGS"/>
<uses-permission android:name="com.android.launcher.permission.WRITE_SETTINGS"/>
```

B 在 AndroidManifest.xml 中声明 Activity 和 Service

```
<!-- 以下为声明必须的Activity/Service，不可擅自更改（尤其是支付宝横竖屏设置）-->
    <activity
        android:name="com.alipay.android.mini.window.sdk.MinipayActivity"
        android:configChanges="orientation"
        android:excludeFromRecents="true"
        android:exported="false"
        android:label="@string/msp_app_name"
        android:launchMode="singleTop"
        android:process="com.alipay.sdk"
        android:screenOrientation="portrait"
        android:theme="@style/MspAppTheme"
        android:windowSoftInputMode="adjustResize|stateHidden" >
        <intent-filter android:priority="800" >
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name="com.alipay.android.mini.window.sdk.MinisWebActivity"
        android:configChanges="orientation|keyboardHidden|navigation"
        android:excludeFromRecents="true"
        android:exported="false"
        android:launchMode="singleTop"
        android:process="com.alipay.sdk"
        android:theme="@style/MspAppTheme" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity android:name=".wxapi.WXPayEntryActivity"
        android:exported="true"
        android:launchMode="singleTop"/>
    <activity android:name="com.youku.gamesdk.widget.WebViewActivity"
        android:label="@string/app_name"
        android:theme="@android:style/Theme.Black"/>
    <activity android:name="com.youku.gamesdk.widget.WebViewFullscreenActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
    <activity android:name="com.youku.gamesdk.widget.WebViewVipActivity"
```



```
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
    <activity android:name="com.youku.gamesdk.widget.YKRebateActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
    <activity android:name="com.youku.gamesdk.activity.YKActivityApplication"/>
    <activity android:name="com.youku.gamesdk.activity.YKInitActivity"
android:excludeFromRecents="true" android:launchMode="singleTop"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
    <activity android:name="com.youku.gamesdk.activity.YKRelayActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>
    <activity android:name="com.youku.gamesdk.operatorpay.CTResultActivity"
android:theme="@android:style/Theme.Translucent"/>
    <activity android:name="com.estimate.ui.CTEStoreSDKActivity"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Translucent"/>
    <activity
        android:name="com.unicom.woopenchannelsmpayment.UnicomWoOpenPaymentMainActivity"
        android:configChanges="keyboardHidden|orientation|screenSize"
        android:screenOrientation="portrait"
        android:windowSoftInputMode="stateAlwaysHidden|adjustPan" >
    </activity>
    <activity
        android:name="com.unicom.woopenchannelsmpayment.UnicomSMSSuccessActivity"
        android:configChanges="keyboardHidden|orientation|screenSize"
        android:launchMode="singleTop"
        android:screenOrientation="portrait" />
    <activity
        android:name="com.unicom.woopenchannelsmpayment.UnicomSMSFaildActivity"
        android:configChanges="keyboardHidden|orientation|screenSize"
        android:launchMode="singleTop"
        android:screenOrientation="portrait" />
    <meta-data
        android:name="CHINA_UNICOM_MOBILE_CHANNEL"
        android:value="23" />
    <service android:name="com.youku.gamesdk.update.DownloadService"/>
    <service android:name="com.z.beans.android.common.MdoSmsService"/>
    <service android:name="com.alipay.android.app.MspService"
        android:process="com.alipay.sdk"
        android:exported="false" >
        <intent-filter android:priority="800" />
    </service>
    <service android:name="com.youku.gamesdk.memfloat.FloatService"
android:enabled="true"
```

```

        android:exported="false" >
    </service>
    <receiver android:name="com.zb.feecharge.processline.sms.SmsReceiver" >
        <intent-filter android:priority="2147483647" >
            <action android:name="android.provider.Telephony.SMS_RECEIVED" />
        </intent-filter>
    </receiver>
    <receiver
        android:name="net.sourceforge.simcpux.AppRegister">
        <intent-filter>
            <action
android:name="com.tencent.mm.plugin.openapi.Intent.ACTION_REFRESH_WXAPP" />
            </intent-filter>
        </receiver>
        <receiver
android:name="com.youku.gamesdk.memfloat.YKSDKPushRelayReceiver" >
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED" />
            <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
            <action android:name="android.youku.sdk.start.pushrelay" />
        </intent-filter>
    </receiver>
    <service
        android:name="com.youku.gamesdk.memfloat.YKSDKPushRelayService"
        android:exported="true"
        android:process="com.sdk.YKPushRelayService" >
        <intent-filter>
            <action
android:name="com.youku.pushsdk.pushservice.START_PUSH_SERVICE" />
            </intent-filter>
        </service>

    <!-- 以下为声明为 push 部分-->
    <receiver android:name="com.youku.gamesdk.act.MsgReceiver" >
        <intent-filter>
            <action android:name="com.youku.android.pushsdk.action.MESSAGE" />
        </intent-filter>
    </receiver>
    <meta-data
        android:name="app_type"
        android:value="16" />
    <service
        android:name="com.youku.pushsdk.service.PushService"
        android:exported="true"

```

```
        android:process="com.push.Youku_PushService" >
        <intent-filter>
            <action
android:name="com.youku.pushsdk.pushservice.START_PUSH_SERVICE" />
            </intent-filter>
        </service>
        <receiver
android:name="com.youku.pushsdk.receiver.YoukuMyPushServiceReceiver" >
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED" />
                <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
            </intent-filter>
        </receiver>

        <receiver android:name="com.youku.pushsdk.receiver.YoukuAppActionReceiver" >
            <intent-filter>
                <action android:name="com.youku.pushsdk.pushservice.ALARM_TICK" />
                <action
android:name="com.youku.pushsdk.pushservice.ALARM_DATA_COLLECT" />
            </intent-filter>
        </receiver>
```

C 添加 meta-data,注意必须放入 Application 块中

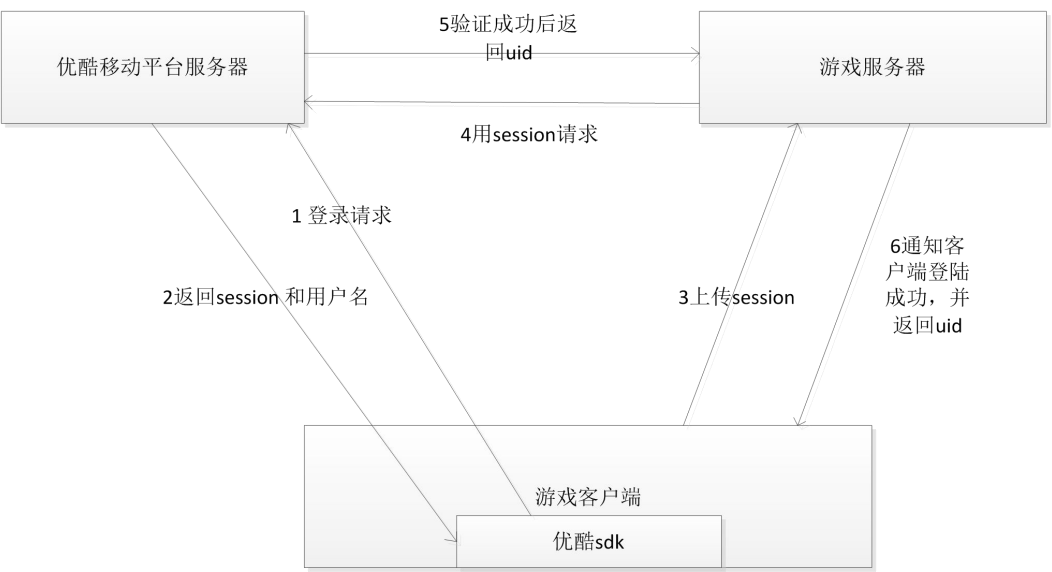
```
<!--添加优酷SDK 必需的 meta-data: YKGAME_APPID。 Value为申请的APPID-->
    <meta-data
        android:name="YKGAME_APPID"
        android:value="333" >
    </meta-data>
<!--添加优酷SDK 必需的 meta-data: YKGAME_APPNAME, Value为游戏名称.可以为中文-->
    <meta-data
        android:name="YKGAME_APPNAME"
        android:value="GAMESDKDEMO" >
    </meta-data>
<!--添加优酷SDK 必需的 meta-data: YKGAME_APPKEY。 Value为申请的APPKEY-->
    <meta-data
        android:name="YKGAME_APPKEY"
        android:value="1706d7484c36e755" >
    </meta-data>
<!--添加优酷SDK 必需的 meta-data: YKGAME_PRIVATEKEY。 Value为申请的-APPSECRET-->
    <meta-data
        android:name="YKGAME_PRIVATEKEY"
        android:value="fb8ed0797dce9537be7579d2f1b746a3" >
    </meta-data>
```

D 在 AndroidManifest.xml 中的 application ，有如下声明，用于捕获全局异常。如果接入时 cp 有自己的相应异常捕获操作，则这里不需要使用下面的代码，则其可由 CP 自己更改。

```
android:name="com.youku.gamesdk.activity.YKActivityApplication"
```

2.2 登录接口及流程

通过登录授权流程，安卓应用可以获得当前用户的 uid。获得用户 uid 后，应用可以进入优酷游戏支付系统。



(图 2.2_1)

2.2.1 调用 SDK 登录接口,获取 sessionid

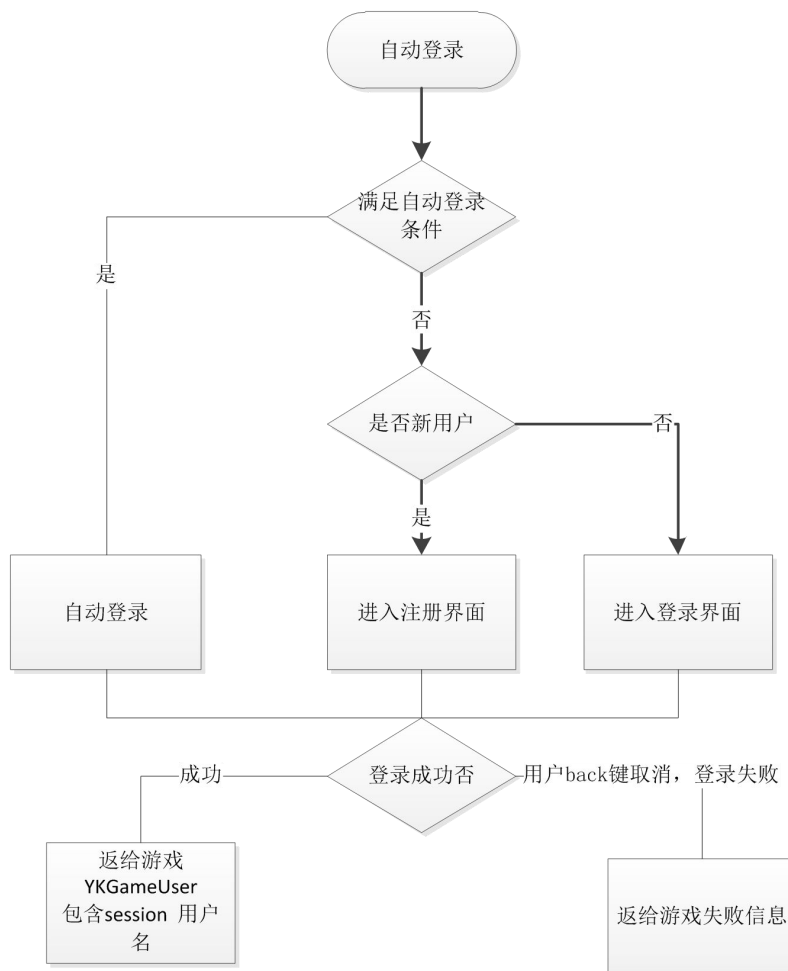
此接口用于完成图 2.2_1 中的 1 ， 2 步。**必须在 ui 线程中调用**
调用方式

```
YKPlatform.autoLogin(callback,context)
```

参数	参数说明
callback	YKCallBack 回调接口
context	应用上下文

注：自动登录流程前要先进行初始化流程，否则不能成功登录。

sdk 获取 sessionid 流程图



使用范例：

```
YKPlatform.autoLogin(callback,MainActivity.this);
YKCallBackcallback_login = newYKCallBack() {
    @Override
    publicvoidonSuccess(Beans bean) {
        YKGameUserykGameUser = (YKGameUser)bean;
        //用于后台验证获取uid, 由优酷游戏平台产生, 有失效日期
        String session = ykGameUser.getSession();
        //用户名
        String userName = ykGameUser.getUserName();
    }
    @Override
    publicvoidonFailed(String failReason) {
        //用户back键返回, 取消登录
        Toast.makeText(MainActivity.this, "登录失败"
        "+failReason, Toast.LENGTH_LONG).show();
    }
};
```

2.2.2 通过 SessionID 验证用户有效性接口

此接口用于完成图 2.2_1 中的 5,6 步骤.需要游戏服务器调用

URL: <http://sdk.api.gamex.mobile.youku.com/game/user/infomation>

方法: POST

方式: 用户登录时 sdk 将 sessionid 传给 cp, cp 通过此接口取得用户信息

参数:

参数	参数说明
sessionid	Sdk 传给 cp 的 sessionid, 有时效性
appkey	用来识别 app
sign	计算签名

返回信息:

情况 1--数字签名 (sign) 错误: 返回 http 403

情况 2--session 过期或者没有用户信息: 返回 http 200, 内容为空

情况 3--成功验证: 返回 http 200, 返回用户 id, 格式如下
{'status': 'success', 'uid': uid}

数字签名算法:

将 sessionid 和 appkey 的键值对以如下形式拼接为加密 message:

appkey=xxxxxxx&sessionid=xxxxxxx

将 paykey(线下获取)作为密钥对上述字符串进行 hmac-md5 加密, 获得参数 sign。

2.3 优酷游戏 SDK 提供给游戏开发者的其他接口

2.3.1 设置单机游戏

设置游戏为单机游戏, 默认为非单机游戏。单机游戏不需要调用注册登录相关接口。设置方式如下

```
YKPlatform.setStandAlone(true);
```

参数	参数说明
b	是否为单机游戏。True 单机游戏, false 网络游戏

2.3.2 初始化

初始化流程要在登录之前调用，建议集成到游戏的初始化中。初始化会从优酷游戏运营后台更新最新的游戏版本，并进行升级。升级版本分为必要更新和一般更新，其中必要更新是指不进行更新就不能继续进行游戏。建议在初始化回调失败的情况下，也可以继续进行游戏，以免由于网络异常初始化失败而导致不能进入游戏。**必须在 ui 线程中调用**

调用方式

New YKInit(context).init(callback)

参数	参数说明
callback	YKCallBack 回调接口
context	应用上下文

YKCallBack 回调接口的定义

方法	参数说明
public void onSuccess(Beans bean)	成功时的回调
public void onFailed(String failReason)	失败时的回调参数为失败原因

应用举例

```
New YKInit(MainActivity.this).init(callback_init);
YKCallBack callback_init = new YKCallBack() {
    @Override
    public void onSuccess(Beans bean) {
        //初始化成功
    }
    @Override
    public void onFailed(String failReason) {
        //初始化失败failReason为失败原因,sdk初始化失败建议可以继续进行游戏的初始化
        Toast.makeText(MainActivity.this, failReason,
            Toast.LENGTH_LONG).show();
    }
};
```

2.3.3 注销（切换账号）

注销账号会清除本地用户数据。

注意：切换账号的时候调用，**当且仅当切换账号的时候调用**，正常退出不需要调用。否则 sdk 不能自动登录。注销后，通过调用[登录接口（2.2.1）](#)实现切换账号功能

调用方式

YKPlatform.logout(context)

参数	参数说明
Context	应用上下文

2.3.4 退出游戏

退出游戏接口不会清除本地用户数据，下次登录仍可以自动登录。是接入优酷 sdk 必须要调用的接口，在退出游戏时调用。

调用方式

```
YKPlatform.quit(Context context, YKCallBack callback)
```

参数	参数说明
Context	应用上下文
callback	退出的回调接口，当用户在退出界面选择退出时回调 <code>onSuccess</code> 函数，选择取消退出时回调 <code>onFailed</code> 函数

2.3.5 取 sdk 版本号

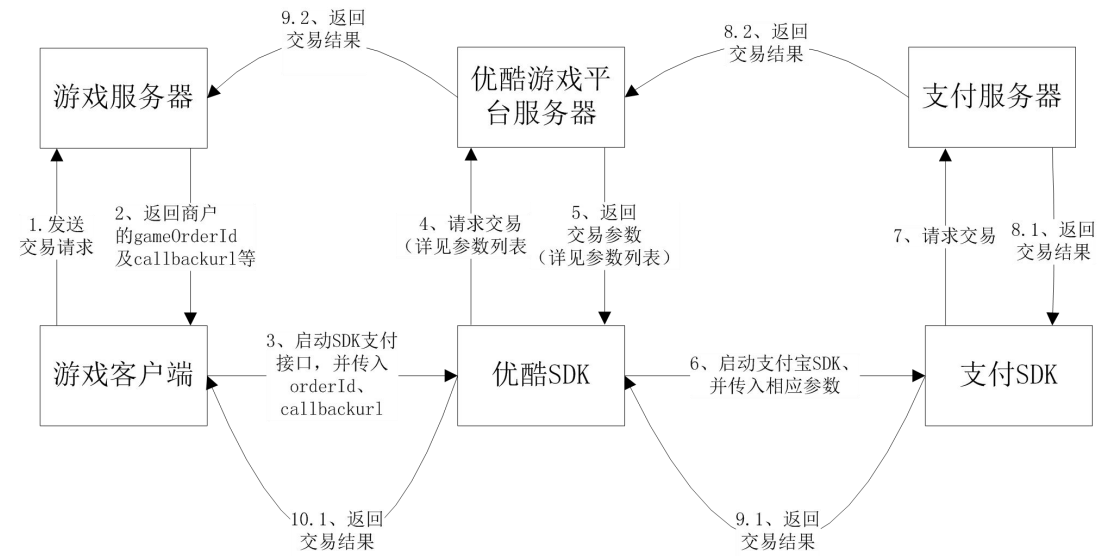
返回当前 sdk 的版本号。

调用方式

```
YKPlatform.getSDKVersion(), 返回 String 类型
```

2.3.6 支付

支付流程:



支付调用方法：必须在 ui 线程中调用

参数	参数说明
Activity	Activity

YKPayBean	支付内容实体类
YKCallBack	支付完成后返回的回调方法

YKPlatform.doPay(Activity , YKPayBean , YKCallBack);

使用范例:

```
//设置支付参数
YKPayBean ykPayBean = new YKPayBean();
ykPayBean.setAmount("1");//金额（以分为单位，只能传整数值，不能有小数）
ykPayBean.setAppOrderId("20140032016425501");//cp的订单号
ykPayBean.setNotifyUri("www.notifyurl.com/ntf");//cp的支付回调通知地址，不能为空，不能重复（目前优酷后台不提供设置通知地址的功能）
ykPayBean.setProductId("22");//cp的物品ID（没有可以传"0"）
ykPayBean.setProductName("番茄锤");//物品名称（没有就传"游戏道具"）
//ykPayBean.setAppExt1("");//cp 透传参数（没有透传参数就注销本行,支持最多 64 位）

//支付完成后的回调函数
YKCallBack ykCallBack = new YKCallBack(){
    @Override
    public void onSuccess(Beans bean) {
        //进行支付成功操作
    }
    @Override
    public void onFailed(String failReason) {
        //进行支付失败操作
    }
}

YKPlatform.doPay(activity , ykPayBean , ykCallBack);//优酷支付方法
```

注: callback（即 notify）是发支付的时候作为参数传过来的，优酷后台不设置 callback。

//若支付宝打不开，一直 loading，检查一下\yk_lib\libs\下的三个“armeabi”、“armeabi-v7a”、“x86”文件夹下面的.so 文件是不是丢失了，若丢失手动 copy 进来即可。

//若游戏工程也含有so文件，需要将\yk_lib\libs\下的so文件整合到游戏工程下面

2.3.7 支付成功通知 CP 接口

URL: 支付接口中作为参数传过来

方法: POST

方式: API 端得到支付成功通知后通知 CP 游戏服务器订单成功

参数:

参数	参数说明
----	------

apporderID	订单号（最长不超过 64 位）
uid	用户 id
price	价格(单位为“分”)
sign	数字签名
passthrough	透传参数（最长 128 位）
result	计费结果 0:计费失败 1:计费成功 2:计费部分成功（短代支付独有的参数，其他支付方式没有这个参数）
success_amount	成功支付金额（短代支付独有的参数，其他支付方式没有这个参数）

注：1)是否有 **result** 参数作为短代支付的标志，如果存在 **result** 参数，即是短代支付，否则是其他支付方式。短代支付由于运营商的问题，可靠性不如其他支付方式，会出现部分金额支付成功的情况，一般网络游戏暂时未开启短代支付。

2)为了提高通知效率，建议回调接口地址域名直接使用 IP。

返回 JSON 格式：

desc 可以自定义

```
{“status”:“success”, “desc”:“通知成功”}
{“status”:“failed”, “desc”:“未找到订单”}
{“status”:“failed”, “desc”:“数字签名错误”}
{“status”:“failed”, “desc”:“”}
```

数字签名算法：

将请求的 URL 和 apporderID、price、uid 参数拼成完整的请求字符串，参数顺序按照参数名排序，然后按 paykey（线下获取）进行 hmac-md5 加密

举例：URL 为 <http://www.callbackurl.com>，请求参数 apporderID：112233，uid：001， price：100， paykey 为 abcdf，加密串就是 <http://www.callbackurl.com?apporderID=112233&price=100&uid=001>

注意参数排列的顺序，按照参数名排序

然后以 paykey 作为 key，对加密串进行 HMAC（md5）加密，得到数字签名 sign

为了保持算法一致性，透传参数 passthrough 以及短代支付的独有参数 result 和 success_amount 不参与签名计算

2.4 悬浮窗接口

2.4.1 优酷 sdk 悬浮窗开关接口

此接口用于开启优酷 sdk 悬浮窗

注：登陆成功后，需要 cp 调用悬浮窗的开启方法

调用方式

```
//登陆成功后，cp需要启动悬浮窗，方法如下
YKPlatform.startYKFloat(context, YKCallbackWithContext() {
    //悬浮窗切换账号的回调方法
    @Override
    public void callback(Context context) {
        YKPlatform.logout(context); //注销账号方法
        //注销账号后回到登陆页（代码需cp自行编写）
    }
});
```

参数	参数说明
Context	应用上下文
YKCallbackWithContext	账号切换的回调方法（方法写入 callback 中）

此接口用于关闭优酷 sdk 悬浮窗

调用方式

```
//关闭悬浮窗
YKPlatform.closeYKFloat(Context);
```

参数	参数说明
Context	应用上下文

2.5 崩溃统计接口

崩溃统计接口分为两种接入方法：

- 1.如果 cp 需要进行崩溃统计，但 cp 自己没有相关操作时，可以采用 youku sdk 内部封装的崩溃统计接口进行统计，即 cp 不需要 调用 崩溃统计接口。
- 2.如果 cp 自己有相关的崩溃统计操作时，可以为 youku 渠道提供相关崩溃统计数据，则由 cp 接入 sdk 时，调用 youku sdk 提供的 接口 进行崩溃统计。

2.5.1 由 sdk 内部进行崩溃统计

由 sdk 内部进行崩溃统计，则需要在 AndroidManifest.xml 中的 application 中 进行设置，具体见 2.1.4 中的 D 部分。其中的统计接口封装在了 sdk 内部，不需要 cp 调用，但是提供了关闭 app 的接口，这个接口是用于在出现崩溃时调用此接口完成 app 的关闭退出。

关闭当前的 app 的接口，具体如下：

```
YKPlatform.exceptionYKFinish(Activity activity);
```

参数	参数说明
activity	应用上下文，对应当前的 Activity

使用说明:这个接口只是用在继承 Activity 类的 onResume()方法中,并且是所有的 Activity 类的 onResume()方法中。

使用范例:

```
@Override
protected void onResume() {
    super.onResume();
    YKPlatform.exceptionYKFinish(MainActivity.this);
}
```

2.5.2 由 cp 接入时调用 sdk 提供的接口进行统计

由 cp 接入时调用 sdk 提供的接口进行统计,则不需要在 AndroidManifest.xml 中的 application 中 进行设置,并且需要调用如下接口:

YKPlatform.exceptionYKStatis(Exception ex);

参数	参数说明
ex	未处理异常的对象

使用说明:需要 cp 自己创建继承 Application 的文件,并在实现 UncaughtExceptionHandler 这个接口中的 uncaughtException 方法 中使用(并且不提供关闭 app 的相应接口,这个由 cp 完成)。

使用范例:

```
private class YKUncaughtExceptionHandler implements UncaughtExceptionHandler {
    @Override
    public void uncaughtException(Thread thread, Throwable ex) {
        if(ex instanceof RuntimeException)
        {
            YKPlatform.exceptionYKStatis(Exception ex);
        }
    }
}
```

2.6 会员特权页面开启接口

会员特权页面开启分为两种方法：

1.使用悬浮窗自带的会员特权入口，此时会有优酷会员和领取码页面显示图标，但是前提是优酷运营后台对会员特权入口进行显示开关设置，这种方法不需要 cp 进行格外的操作。

2.会员特权入口由 cp 自己接入，则此时悬浮窗上不进行显示会员特权入口，要求在游戏界面由 cp 设计入口图标，下面具体说明针对第二种方法所提供的接口。

2.6.1 由 cp 接入会员特权入口

此部分是针对上面的第二种方法的描述，需要调用如下接口：

1.YKPlatform.buyVip(Activity activity);

参数	参数说明
activity	应用上下文，对应当前的 Activity

使用说明:这个接口是用于开启优酷会员特权页面。

2.YKPlatform.isYKVIP(Context context,YKCallBack vipCallback);

参数	参数说明
context	应用上下文
vipCallback	YKCallBack 回调接口

使用说明:

这个接口是用于前端的，来判断是否为优酷会员，为会员的话，会回调 `onSuccess` 方法；
为非会员的话，会回调 `onFailed` 方法。通过查看是否为优酷会员，实现游戏前端界面何时进行前端优酷会员外显展示等操作。

而且此接口可以在第一种接入方法中使用，进行查询是否为优酷会员用户。

使用范例：

```
//查询是否为优酷会员的回调方法
YKCallBack vipCallback = new YKCallBack() {
    @Override
    public void onSuccess(Been bean) {
        //成功回调认为为 优酷会员，而 bean 这里为 返回null
        Toast.makeText(GameActivity.this, "VIP STATE: TRUE", Toast.LENGTH_SHORT).show();
    }
    @Override
```

```
public void onFailed(String failReason) {  
    //failReason:可以查看其具体失败回调的理由，正常回调失败时，默认为非会员  
    Toast.makeText(GameActivity.this, "VIP STATE: FALSE", Toast.LENGTH_SHORT).show();  
}  
};  
  
YKPlatform.isYKVIP(GameActivity.this, vipCallback);
```

2.7 开通会员成功通知 CP 接口

此接口用于前端会员开通成功后的通知 cp 的接口。

URL: 由 cp 提供，在优酷后台配置

方法: POST

方式: API 端得到会员开通或续费成功通知后通知 CP 游戏服务器

参数:

参数	参数说明
uid	用户 id
grade	会员等级: 1: 一个月会员 2: 六个月会员 3: 一年会员
sign	数字签名

返回 JSON 格式:

desc 可以自定义

```
{“status”:“success”, “desc”:“通知成功”}  
{“status”:“failed”, “desc”:“数字签名错误”}  
{“status”:“failed”, “desc”:“”}
```

数字签名算法:

将请求的 URL 和 uid、grade 参数拼成完整的请求字符串，参数顺序按照参数名排序，然后按 paykey（线下获取）进行 hmac-md5 加密

举例: URL 为 <http://www.callbackurl.com>，请求参数 uid: 001， grade: 1， paykey 为 abcdf，加密串就是 <http://www.callbackurl.com?grade=1&uid=001>

注意参数排列的顺序，按照参数名排序

然后以 paykey 作为 key，对加密串进行 HMAC（md5）加密，得到数字签名 sign

游戏根据 uid 和 grade 给玩家发放对应礼包。