

YSSDK 开发文档

SDK 开发文档.....	1
1.1 本案目的.....	1
1.2 集成 SDK.....	1
1.2.1 引进资源包.....	2
1.2.2 配置工程 AndroidManifest.xml	5
1.3 对接登录系统.....	9
1.3.1 登录流程图.....	9
1.3.2 登录接口说明.....	9
1.4 对接充值系统.....	12
1.4.1 充值流程图.....	12
1.4.2 充值接口.....	12
1.5 浮点功能.....	14
1.6 资源回收接口(游戏退出必须调用)	15
1.7 充值服务器回调接口开发.....	15
1.7.1 通知充值结果.....	15
1.7.2 签名说明.....	16
1.7.3 合作商响应请求.....	16
1.7.4 支付编码表.....	16
1.7.5 温馨提示.....	17
1.8 运行环境.....	17
1.9 混淆.....	17

1.1 本案目的

本文档提供给游戏合作商对接 SDK 相关人员使用，文档主要分四大部分，一：集成 SDK；二：对接登录系统；三：对接充值系统；四：充值服务器回调接口

定义简称：游戏合作商：统称“合作商”；运营方：“YS”

1.2 集成 SDK

调用 yssdk.jar 及相关、配置清单 AndroidManifest.xml

1.2.1 引进资源包

引进 jar 包是基于 eclipse 开发 IDE 上实现，具体流程如下：

1：引入相关 jar 包和 so 文件以做支持

必要 jar 包引入(在项目中 libs 目录下增加)：

alipaySdk-20151112.jar

android-support-v4.jar

commons-codec-1.6.jar

ipaynow_base_v1.3.2.jar

ipaynow_wechatwp_v1.0.1.jar

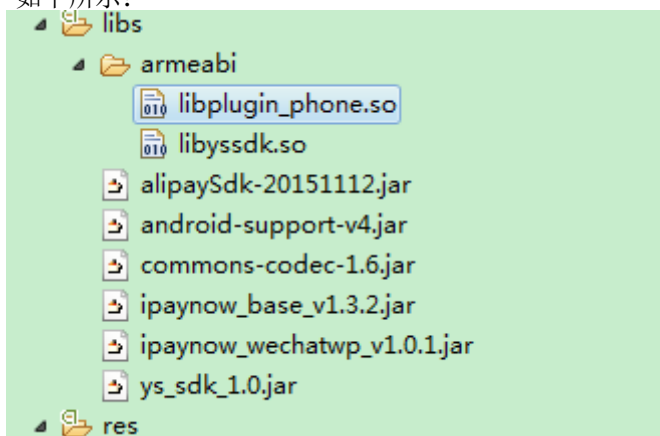
ys_sdk_1.0.jar

必要 so 文件引(在项目中 libs 下 armeabi 文件夹中增加)：

libyssdk.so

libplugin_phone.so

如下所示：



2：引入相关 res 资源文件，res 文件下的资源全部分引入以作显示调用。其中 icon 和样式可以根据需要变更，所有资源 layout 文件名及 id 保持原来命名。如下图所示：



1.2.2 配置工程 AndroidManifest.xml

在工程 AndroidManifest.xml 格式如下,确保工程 AndroidManifest.xml 具有以下权限:
(如有误差, 请以 DEMO 为准)

```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.webkit.permission.PLUGIN" />
<uses-permission
android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"
/>
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission
android:name="android.permission.CHANGE_CONFIGURATION" />
<!-- 在SDCard中创建与删除文件权限 -->
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<!-- 照相的权限 -->
<uses-feature android:name="android.hardware.camera" />

<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.hardware.camera.autofocus" />

<!-- 读取短信验证码 可选 -->
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.READ_SMS" />

<!-- 录像需要的权限 -->
<uses-permission android:name="android.permission.RECORD_VIDEO" />

<!-- floatview -->
<uses-permission
android:name="android.permission.SYSTEM_OVERLAY_WINDOW" />
```

2、在工程 AndroidManifest.xml 的<application>段内配置以下内容：

```
<activity
    android:name="com.game.sdk.ui.SplashActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="sensor"
    android:theme="@android:style/Theme.Light.NoTitleBar.Fullscreen"
/>

<activity
    android:name="com.game.sdk.ui.UpdateActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.ChargeActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.UserCenterActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.YXBRecordActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.PTBRecordActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.MyPropertyActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.NoticeActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.LoginActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
```

```

<!-- 浮点 -->
<activity
    android:name="com.game.sdk.floatwindow.FloatWebActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
<!-- alipay sdk begin -->
<activity
    android:name="com.alipay.sdk.auth.AuthActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>
<activity
    android:name="com.alipay.sdk.app.H5PayActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind"
    android:windowSoftInputMode="adjustResize|stateHidden" >
</activity>
<activity
    android:name="com.game.sdk.ui.AlipayActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.NoDisplay" />
<activity
    android:name="com.game.sdk.ui.NowPayActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.NoDisplay" />
<!-- alipay sdk end -->
<!-- 现在支付 -->
<activity
    android:name="com.ipaynow.plugin.activity.PayMethodActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:exported="false"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Dialog" >
</activity>
<activity
    android:name="com.ipaynow.plugin.inner_plugin.prepay_plugin.activity.PrePayActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:screenOrientation="behind"
    android:windowSoftInputMode="adjustResize" >
</activity>
<activity
    android:name="com.ipaynow.plugin.inner_plugin.prepay_plugin.activity.NotifyActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>

```

```
<!-- 微信WAP版支付 -->
<activity
android:name="com.ipaynow.plugin.inner_plugin.wechatwp.activity.WeChatNotifyA
ctivity"
android:configChanges="keyboardHidden/navigation/orientation/screenSize"
    android:exported="false"
    android:launchMode="singleTop"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.NoDisplay" />

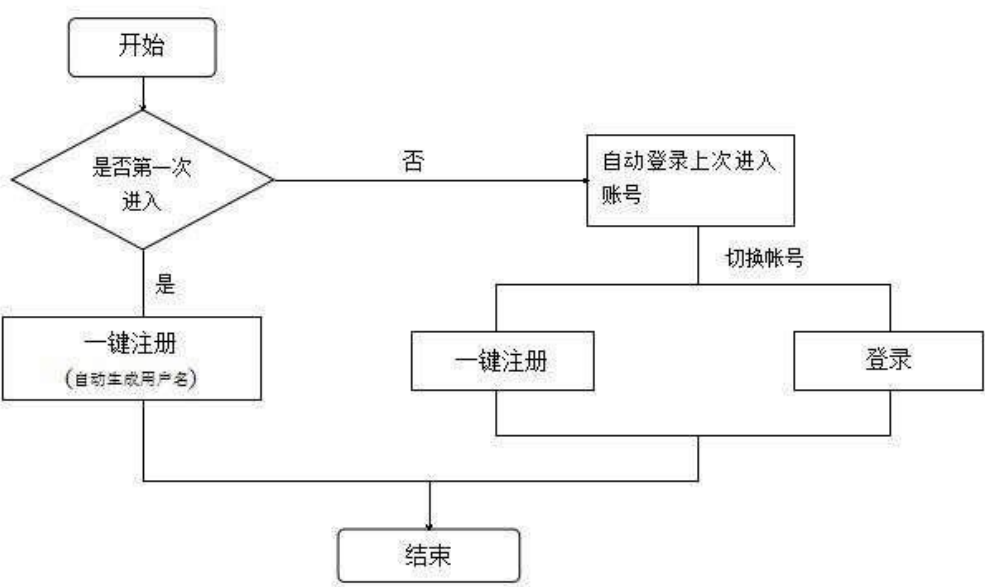
<service android:name="com.game.sdk.SDKService" >
    <intent-filter android:priority="1000" >
        </intent-filter>
    </service>
<!--此处用于配置渠道和 APPID,此处参数由 YS 提供-->
<meta-data android:name="APPID"android:value="1"/>
<meta-data android:name="GAMEID"android:value="1"/>
<meta-data android:name="AGENT"android:value="default"/>
```

3、在工程 AndroidManifest.xml 的<manifest>段内，设置适应所有分辨率，配置以下内容：

```
<supports-screens
    android:smallScreens="true"
    android:normalScreens="true"
    android:largeScreens="true"
    android:resizeable="true"
    android:anyDensity="true"/>
```

1.3 对接登录系统

1.3.1 登录流程图



1.3.2 登录接口说明

合作商调用相关接口获取登录数据后，需要做验签处理，具体如下：

1. 调用 SDKManager 的 `getInstance(Context ctx)` 静态方法来获取 SDKManager 的实例(保险起见 `getInstance(Contextctx)`可以多次调用，不会影响程序运行性能)。

调用 SDKManager 的 `public void showLogin(context, boolean isShowQuikLogin, OnLoginListenerloginlistener)`方法显示登录界面

参数名	类型	参数说明
<code>context</code>	<code>context</code>	上下文 context 实例
<code>isShowQuikLogin</code>	<code>boolean</code>	是否显示快捷登录， <code>true</code> 表示显示快捷登录， <code>false</code> 表示直接显示登录界面

`onloginListener` `OnLoginListener` 登录接口的回调，示例如下

```
sdmanager.showLogin(context,true,new OnLoginListener(){
    //成功登录回调
    @Override
    public void loginSuccess(LoginCallback
        logincallback){
        //登录成功返回的签名，需要做验证处理
        String sign=logincallback.sign;
        //登录回调时间戳
        Long logintime=logincallback.logintime;
        //KD 登录的用户名
        String username=logincallback.username;
    }
    //登录失败回调可能是用户名密码不正确，也有可能是 KD 服务端出现了问题
    @Override
    public void loginError(LoginErrorMsg errorMsg){
        //登录失败错 误码
        int code=errorMsg.code;
        //登录失败的消息 提示
        String msg=errorMsg.msg;
    }
});
```

成功登录的信息体 `LoginCallback` 解析

参数名	类型	参数说明
<code>username</code>	<code>String</code>	登录成功后，用户的用户名（唯一）
<code>logintime</code>	<code>long</code>	用户登录的时间戳 用来登录验签比
<code>sign</code>	<code>String</code>	(签名生成规则)

登录失败信息体 `LoginErrorMsg` 解析

参数名	类型	参数说明
<code>code</code>	<code>int</code>	登录失败错误码[-1 账号为空， -2 密码为空 -3 注册设备来源为空 -4 平台应用 key 不对 -5 渠道 ID 为空 -6 服务器内部错误 0 为未知错误]
<code>msg</code>	<code>String</code>	登录失败的消息提示

参数签名规则

参数名	类型	参数说明	签名顺序
username	String	KD 用户帐号	1
appkey	String	应用 appkey,由 YS 提供	2
logintime	String	登陆时间, 时间戳格式	3

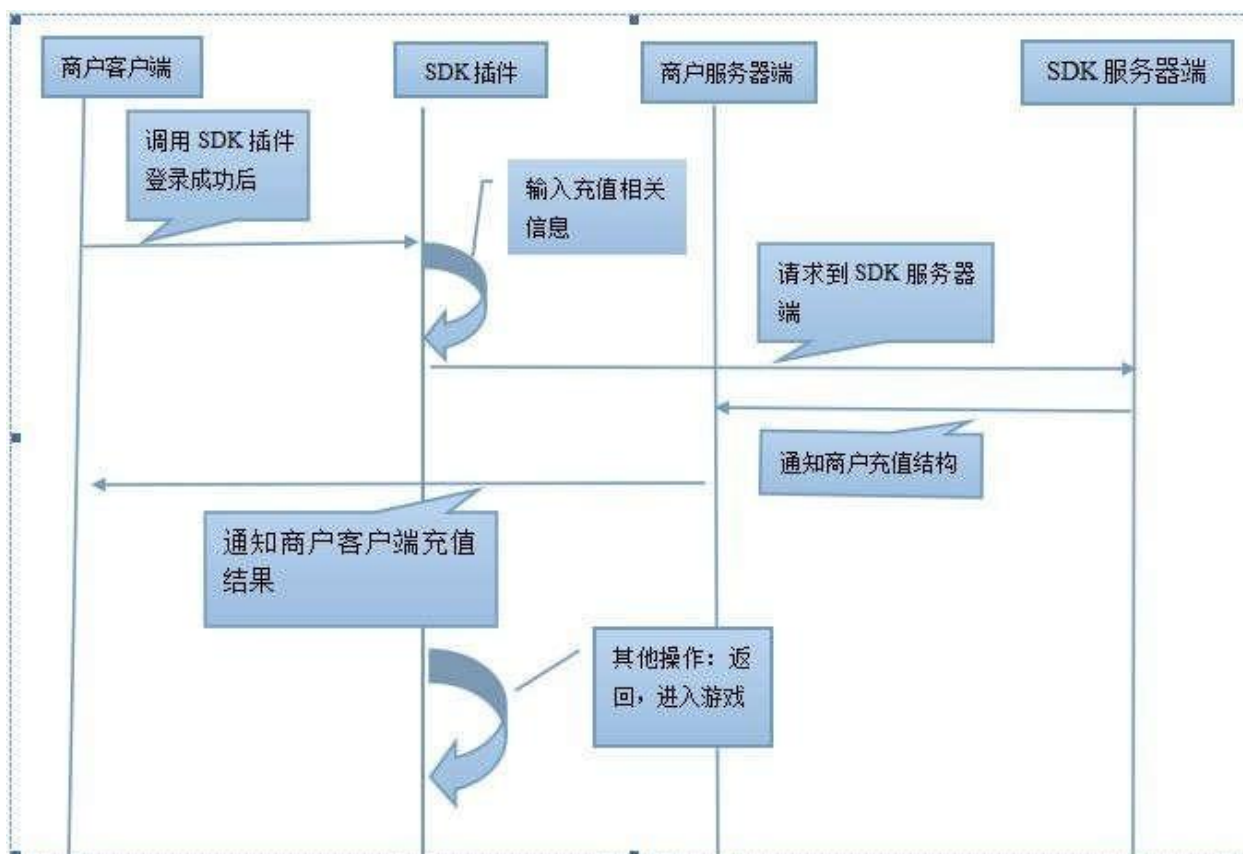
合作商可根据登录成功返回的信息, 跟自己的服务端做 SIGN 二次验证处理, sign 签名如下 (不需要进行 urlencode) :

签名字符串示例:

```
sign=MD5("username=t315688&appkey=91bac46a9b70bd2db563cc483d443ba3&logintime=1395634100")
```

1.4 对接充值系统

1.4.1 充值流程图



1.4.2 充值接口

1. 调用 SDKManager 的 getInstance(Context ctx) 静态方法来获取 SDKManager 的实例。
2. 调用 SDKManager 的

// 构造方法

```
public void showPay(context,roleid,money, serverid, productname,  
productdesc, attach,paymentListener)
```

充值界面。参数说明：

参数名	类型	是否必须	参数说明
context	Context	是	上下文，
roleid	String	是	调用者的实例角色 id
money	String	是	充值金额(分)(整数，不要包含小数
serverid	String	是	合作商服务器 id
productname	String	是	产品名字(游戏名称)
productdesc	String	是	产品描述
Attach	String	是	扩展参数(若有自定义参数传递)
paymentListener	OnPaymentListener	是	对充值是否成功进行监听

充值示例如下：

```
sdkmanager.showPay(context,  
roleid, money,  
serverid,  
productname,  
productdesc,  
attach,  
new OnPaymentListener(){  
    @Override
```

```

    public void paymentSuccess(PaymentCallbackInfo paymentCallbackInfo)
    {
        //ys 用户充值的金额数
        int money=paymentCallbackInfo.money;
        //附加消息
        String msg=paymentCallbackInfo.msg;
    }
    @Override
    public void paymentError(PaymentErrorMsg errorMsg)
    {
        //返回的错误代码
        int code=errorMsg.code;
        //意图充值的金额
        int money=errorMsg.money;
        //附加消息
        String msg=errorMsg.msg;
    }
});

```

充值成功信息体 **PaymentCallbackInfo** 解析

参数名	类型	参数说明
msg	String	充值结果描述
money	int	充值金额数，单位（分）

充值失败信息体 **PaymentCallbackInfo** 解析

参数名	类型	参数说明
code	String	状态码
money	int	意图充值金额数，单位（分）
msg	String	充值失败消息提示

1.5 浮点功能

函数：

```

public void showFloatView()//显示悬浮窗口

public void removeFloatView()//隐藏悬浮窗口

```

1. 登陆成功后的回调方法中调用 showFloatView 方法，显示悬浮窗口。按下面设置：

```

|
sdkmanager.showLogin(this, true, new OnLoginListener() {
    @Override
    public void loginSuccess(LoginCallback logincallback) {
        sdkmanager.showFloatView();//显示浮点
    }

    @Override
    public void loginError(LoginErrorMsg errorMsg) {
        // TODO Auto-generated method stub
    }
});

```

- 2.在 Activity.onStop()方法中调用 removeFloatView()方法，保证中断游戏，浮点隐藏。在 Activity.onResume()方法中调用 showFloatView()方法，保证回到游戏后，浮点显示，如下方式：

```

@Override
protected void onStop(){
    // TODO Auto-generated methodstub
    sdkmanager.removeFloatView();//隐藏悬浮窗口
    super.onStop();
}
@Override
protected void onResume(){
    // TODO Auto-generated methodstub
    sdkmanager.showFloatView();//显示悬浮窗口
    super.onResume();
}

```

1.6 资源回收接口(游戏退出必须调用)

- 1.调用 SDKManager 的 getInstance(Context ctx) 静态方法来获取 SDKManager 的实例。
- 2.调用 SDKManager 的 recycle() 方法进行资源回收。

注：该方法在游戏退出时调用(必须要调用)

如：

```

@Override
protected void onDestroy(){
    // TODO Auto-generated methodstub
    sdkmanager.recycle();//退出调用
    super.onDestroy();
}

```

1.7 充值服务器回调接口开发

合作商需要开发一个接收充值结果的系统，提供该系统的 URL 给 YSSDK，由 YSSDK 发起请求，合作商响应请求。请求协议采用 **httppost**

1.7.1 通知充值结果

玩家在游戏中使用 YS 手游 SDK 进行充值，YS 在处理完充值流程之后，主动请求通知游戏方，游戏方未返回接收成功（接收成功判断请参考[合作商响应请求](#)），YSSDK 会启动系统重发机制(一共发送五次，如果五次没有响应将由工作人员补单)。注意只有充值成功才会回调通知结果，充值失败的不会通知。

合作商接收到请求后，需要对所有 String 类型的参数值做 urldecode 处理，urldecode 编码统一为 UTF-8，请求参数具体字段说明如下：

参数名	类型	长度	参数说明	签名顺序
orderid	String	35	sdk 订单号	1
username	String	30	sdk 登录账号	2
gameid	int	11	游戏 ID	3
roleid	String	30	游戏角色 ID	4
serverid	int	11	服务器 ID	5
paytype	String	10	支付类型,支付参数说明	6
amount	float	11	成功充值金额，单位(元)	7
paytime	int	11	玩家充值时间，时间戳形式，如 1394087000	8
attach	String		商户拓展参数	9
sign	String	32	参数签名（用于验签对比）	10

1.7.2 签名说明

合作商接收到 SDK 的请求后，需要获取相关参数并做验签处理，验签规则如下：以[请求参数字段说明](#)中的签名顺序签名，规则为 key=value 的形式，参数之间以“&”符号相连，组成一串字符串。目前包共 10 个签名字段。

签名算法为 MD5，统一使用 32 位 UTF-8 加密算法，编码后转为小写与参数中的 sign 进行对比，如果相同-则签名通过，否则-失败。

签名字符串示例：

```
sign
=MD5("orderid=urlencode(100000)&username= urlencode(zhangsan)&gameid=
urlencode(6)&roleid= urlencode(zhangsanfeng)&serverid= urlencode(1)&
paytype= urlencode(1)&amount= urlencode(1)&paytime= urlencode(20130101125612)&attach=
urlencode(test)&appkey= urlencode(123123123213)")
```

注意：验签参数值为 urlencode 后的内容，如果参数没有数据值，请以“key=”的形式进行签名，例如：paytime=&attach=自定义。要注意字段的大小写哦

appkey由yssdk提供，登录、充值使用同一个appkey

1.7.3 合作商响应请求

合作商接收到 SDK 发出的请求后，以纯字符串的形式返回下列代码，
例如
`out.print("success");`

代码	代码描述
success	接收成功
errorSign	签名错误
error	未知错误

订单状态为失败的也需要返回"success"，如果合作商未返回"success"，YS会启用系统重发机制

1.7.4 支付编码表

支付编码	支付类型描述
zfb	支付宝快捷支付
weixin	微信支付
money	余额支付

订单状态为失败的也需要返回"success"，如果合作商未返回"success"，YS会启用系统重发机制

1.7.5 温馨提示

1. 由于 SDK 的同步系统暂时并不能绝对保证一个订单只同步一次(订单本身是唯一的)，因此建议合作商在发放游戏虚拟货币的时候做一次订单的判断，避免同一订单多次发放
2. 总结运营阶段的经验，SDK 发现支付合作商有时候一个订单会同步两次状态，第一次为失败，第二次成功（实际为成功）。因此建议合作商接收 SDK 同步的订单时处理一下此类情况，例如：系统考虑支持同一个订单，第一次同步失败状态，第二次同步成功状态，系统照常发放游戏币。

1.8 运行环境

1. 目前该 sdk 只支持 Android 2.2 或以上版本的手机系统
2. 手机必须处于网络环境中，2.5G/3G/WIFI 等等

1.9 混淆

YSSDK 包是以 jar 包及资源文件提供给用户的，其中 jar 包已经半混淆状态，您在混淆自己 APK 包的时候请不要将 YSSDK 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：
`-libraryjarslibs/*.jar` 以避免 YSSDK 的相关的 jar 包被混淆。