**Wancms SDK 开发文档**

编写：曹阳

**2018年6月**

**Wancms SDK 开发文档**

目录

[1.1 编写目的 3](#_Toc517945152)

[1.2集成 SDK 3](#_Toc517945153)

[1.2.1引进WancmsSDKLib 3](#_Toc517945154)

[1.2.2配置工程AndroidManifest.xml 5](#_Toc517945155)

[1.3 对接登录系统 8](#_Toc517945156)

[1.3.1 登录流程图 8](#_Toc517945157)

[1.3.2 登录接口说明 8](#_Toc517945158)

[1.4 对接充值系统 10](#_Toc517945159)

[1.4.1 充值流程图 10](#_Toc517945160)

[1.4.2 充值接口 10](#_Toc517945161)

[1.5设置角色基本数据 12](#_Toc517945162)

[1.6 浮点功能 13](#_Toc517945163)

[1.7 资源回收/用户登出接口(游戏退出必须调用) 13](#_Toc517945164)

[1.8 充值服务器回调接口开发 14](#_Toc517945165)

[1.8.1 通知充值结果 14](#_Toc517945166)

[1.8.2 签名说明 14](#_Toc517945167)

[1.8.3 合作商响应请求 15](#_Toc517945168)

[1.8.4 支付编码表 15](#_Toc517945169)

[1.8.5 温馨提示 15](#_Toc517945170)

[1.9 服务器端强制用户下线（可选） 16](#_Toc517945171)

[1.10 服务器端的登陆二次验证 16](#_Toc517945172)

[2.0 运行环境 17](#_Toc517945173)

[2.1 混淆 17](#_Toc517945174)

# 1.1 编写目的

本文档提供给游戏合作商对接的同学使用，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

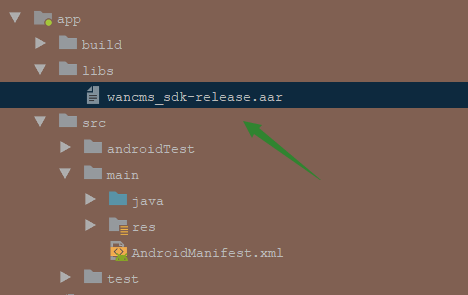
定义简称：游戏合作商：统称“合作商”；运营方：“Wancms”

# 1.2集成 SDK

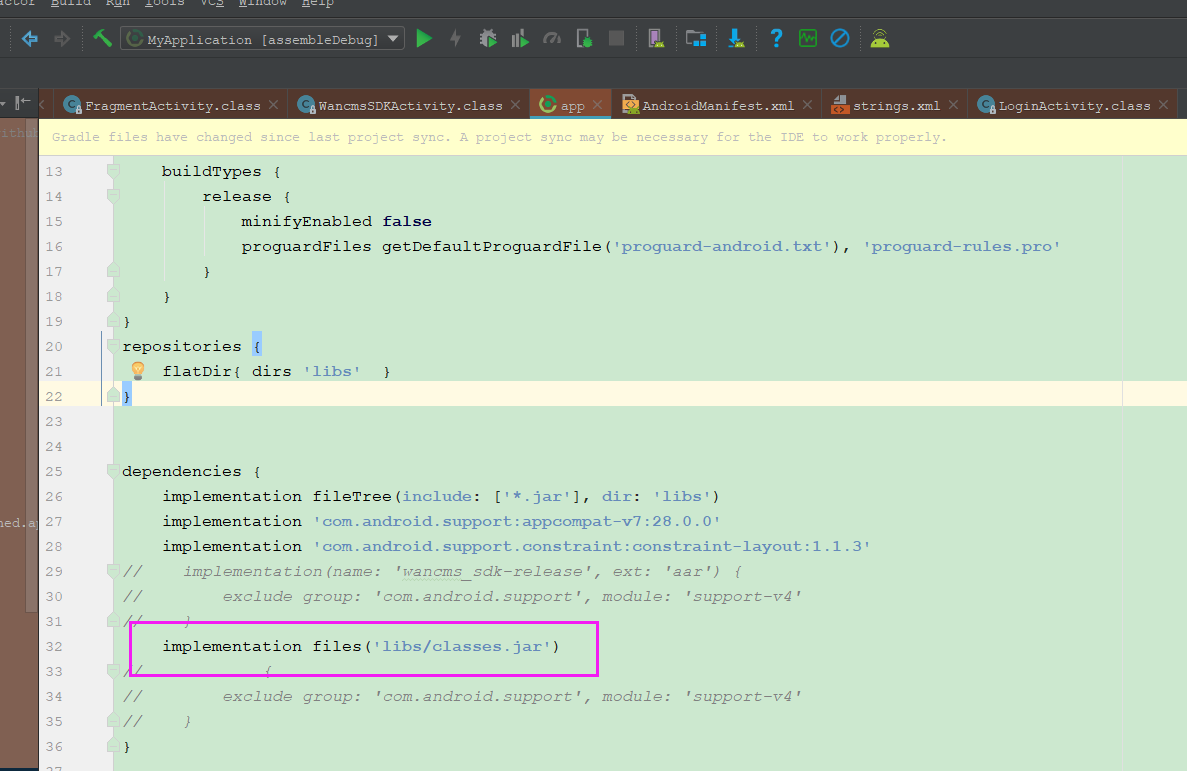
集成出 WancmsSDKLib、配置清单 AndroidManifest.xml

## 1.2.1引进WancmsSDKLib

将jar文件放到对应项目的的lib文件夹下



然后记得gradle里面配置



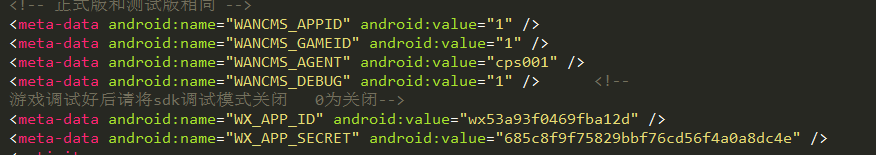
编译通过即可

## 1.2.2配置工程AndroidManifest.xml

**参考demo中的配置。**

**配游戏id,微信等参数:**

运营提供参数



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

注:下面这句话很重要 不要改动上面activity中的这句代码

<supports-screens

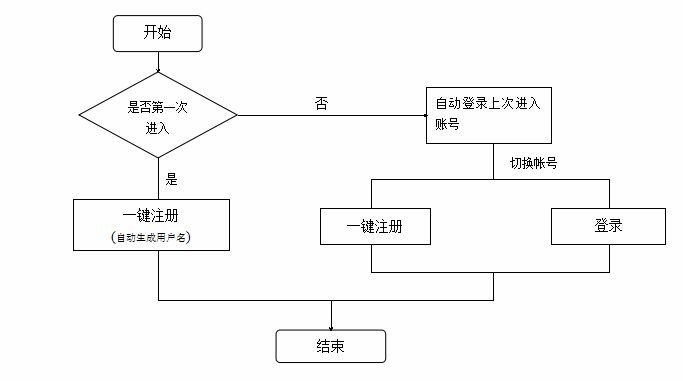
android:smallScreens=*"true"* android:normalScreens=*"true"* android:largeScreens=*"true"* android:resizeable=*"true"* android:anyDensity=*"true"* />

android:configChanges=*"orientation|keyboardHidden|screenSize"*

**1.2.4**在AndroidManifest.xml 的<Appliction>中增加相应的Activity和Service的注册。

# 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 WancmsSDKManager 的 **getInstance**(Context ctx) 静 态 方 法 来 获 取 WancmsSDKManager 的实例(保险起见 **getInstance**(Context ctx)可以多次调用，不 会影响程序运行性能)。

调 用 WancmsManager 的 public void showLogin(Context ctx, boolean isShowQuikLogin,OnLoginListener loginlistener)方法显示登录界面

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| ctx | context | 上下文 context 实例 |
| isShowQuikLogin | boolean | 是否显示快捷登录，true 表示显示快  捷登录，false 表示直接显示登录界面 |
| onloginListener | OnLoginListener | 登录接口的回调，示例如下 |

kdSdk.showLogin(ctx,true,**new** OnLoginListener() {

//成功登录回调

@Override

**public void** loginSuccess(LogincallBack logincallback) {

String sign=logincallback.sign;//登录成功返回的签名，需要做验证

处理

**long** logintime=logincallback.logintime;//登录回调时间戳 String username=logincallback.username;// KD登录的用户名

}

//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) {

**int** code=errorMsg.code;//登录失败错误码 String msg=errorMsg.msg;//登录失败的消息提示

}

});

**成功登录的信息体 LogincallBack 解析**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| **username** | String | 登录成功后，用户的用户名 |
| **logintime** | long | 用户登录的时间戳 |
| **sign** | String | 用来登录验签对比（[签名生成规则](#_bookmark5)） |

**登录失败信息体 LoginErrorMsg 解析**

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
|  |  | 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 |
| **code** | int | -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 |
| **msg** | String | 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 | wancms 用户帐号 | 1 |
| **appkey** | String | 应用 appkey | 2 |
| **logintime** | String | 登陆时间，时间戳格式 | 3 |

签名字符串示例：

sign=MD5(“username=t315688&appkey=91bac46a9b70bd2db563cc483d443ba3&logintime

=1395634100”)

## 1.3.3 登录验证签名规则

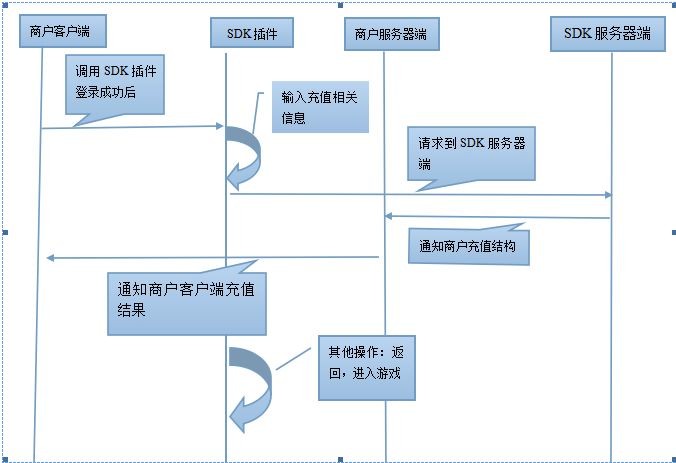
# $str = "&appkey=" . $appkey . "&logintime=" . $logintime . "&username=" . $username;

# 

# $sign = Md5($str);

# 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 WancmsSDKManager 的 getInstance(Context ctx) 静 态 方 法 来 获 取

WancmsSDKManager 的实例。

1. 调用 WancmsSDKManager 的

**public void** showPay(Context ctx**,**String roleid, String money,

String serverid, String productname, String productdesc, String attach,OnPaymentListener paymentListener) 显 示充值界面。参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **是否必须** | **参数说明** |
| ctx | Context | 是 | 调用者的实例 |
| roleid | String | 是 | 角色 id |
| money | String | 是 | 充值金额（元） |
| serverid | String | 是 | 合作商服务器 id (只能传数字 不为0) |
| productname | String | 是 | 产品名字 |
| productdesc | String | 是 | 产品描述 （不能为空字符串） |
| attach | String | 是 | 透传参数（扩展参数） |
| paymentList ener | OnPaymentListen er | 是 | 对充值是否成功进行监听 |

充值示例如下：

kdsdk.showPay(ctx, roleid,

money, serverid, productname, productdesc,

attach**,**

**new** OnPaymentListener() {

@Override

**public void** paymentSuccess(PaymentCallbackInfo paymentCallbackInfo)

{

**int** money=paymentCallbackInfo.money;// KD用户充值的金额数

String msg=paymentCallbackInfo.msg;//附加消息

}

@Override

**public void** paymentError(PaymentErrorMsg errorMsg)

{

}

});

**int** code=errorMsg.code;//返回的错误代码 **int** money=errorMsg.money;//意图充值的金额 String msg=errorMsg.msg;//附加消息

**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 |

**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 |
| msg | String | 充值失败消息提示 |
|  |  |  |

# 1.5设置角色基本数据

调用WancmsSDKManager的 public void setRoleDate(Context ctx,String roleId**,**String roleName,String roleLevel,Int serverId,

String serverName,JSONObject ext)方法设置角色基本数据。

调用时间：在游戏登录验证成功后，进入游戏界面时上报角色信息时调用。

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| ctx | Context | 上下文 context 实例 |
| roleId | String | 角色唯一标识 |
| roleName | String | 角色名称 |
| roleLevel | String | 角色等级 |
| serverId | String | 角色所在区服唯一标识 (只能传数字) |
| serverName | String | 角色所在区服名称 |
| ext | JSONObject | 扩展信息（没有传null） |

# 1.6 浮点功能

**函数：**

public void showFloatView(OnLogoutListener logoutlist)//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。 wancmssdkmanager.showFloatView(onlogoutlistener);

注销回调的声明

**private** OnLogoutListener onlogoutlistener = **new** OnLogoutListener() {

@Override

**public** **void** logoutSuccess(LogoutcallBack logoutncallback) {

}

@Override

**public** **void** logoutError(LogoutErrorMsg errorMsg) {

}

};

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 method stub sdkmanager.removeFloatView();//隐藏悬浮窗口 **super**.onStop();

}

@Override

**protected void** onResume() {

// **TODO** Auto-generated method stub sdkmanager.showFloatView(onlogoutlistener);//显示悬浮窗口 **super**.onResume();

}

3.浮标中的退出登录，用户点击浮标中的注销按钮之后，SDK就会回调showFloatView中注册的退出登录的回调函数，通知到游戏。

# 1.7 资源回收/用户登出接口(游戏退出必须调用)

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

2.调用 WancmsSDKManager 的 recycle()方法进行资源回收。 注：该方法在游戏退出时需要在除OnDestory（）之外的方法里调用(必须要调用)。参考demo

# 1.8 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Wancms，由Wancms发起请求，合作商响应请求。请求协议采用 **http post**

## 1.8.1 通知充值结果

玩家在游戏中使用Wancms手游 SDK 进行充值，Wancms 在处理完充值流程之后，主动请求通知游戏方，游戏方返回接收成功（接收成功判断请参考[合作商响应请求](#_bookmark9)）,Wancms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 | Wancms订单号 | 1 |
| username | String | 30 | Wancms 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](#_bookmark9) | 6 |
| amount | int | 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 | 9 |
| attach | String |  | 商户拓展参数 | 10 |
| sign | String | 32 | 参数签名（用于验签对比） | 11 |

## 1.8.2 签名说明

合作商接收到Wancms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，组成一窜字符串。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=100000&username=zhangsan&gameid=6&roleid=zhangsanfeng& serverid=1&paytype=1&amount=1&paytime=20130101125612&attach=test&appkey=12312 3123213”)

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Wancms提供，登录、充值使用同一个 appkey

## 1.8.3 合作商响应请求

合作商接收到Wancms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”，Wancms会启用系统重发 机制

## 1.8.4 支付编码表

|  |  |
| --- | --- |
| **支付编码** | **支付类型描述** |
| wx | 微信支付 |
| zfb | 支付宝快捷支付 |
| ptb | 平台币支付 |

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

## 1.8.5 温馨提示

1. 由于Wancms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,Wancms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 T Y 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**。

# 1.9 服务器端强制用户下线（可选）

该功能使用于当用户有任何违规操作时可以强制用户下线，合作商需要提供用户强制下线接口连接。接口参数如下：

cp\_url?roleid=roleid&serverid=serverid&message=message&sign=sign

|  |  |  |
| --- | --- | --- |
| **参数名** | **类型** | **参数说明** |
| roleid | String | 角色id |
| serverid | int | 服务器id |
| message | String | 被强制下线原因 |
| sign | String | 签名 |

签名字符串示例：

sign = MD5("roleid=zhangsan&serverid=123456&appkey=对应appid的appkey")；

# 1.10 服务器端的登陆二次验证

请求接口：http://sdkurl/sdkapi/login/verification

请求参数：

username 客户端登陆回调

appid 客户端配置

gameid 客户端配置

logintime 客户端登陆回调

sign 联系平台技术人员获取加密方式

返回形式 通过验证返回 json

{

"msg": "验证不成功",

"status": "2"

}或者

{

"msg": "验证成功",

"status": "1"

}

# 2.0 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

# 2.1 混淆

WancmsSDK 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 KDSDK 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：-libraryjars libs/\*.jar 以避免 KDSDK 的相关的 jar 包被混淆。