

SDK 开发文档

SDK 开发文档.....	1
1.1 编写目的	1
1.2 集成 SDK	1
1.2.1 引进 <i>DM_SdkLib</i>	2
1.2.2 配置工程 <i>AndroidManifest.xml</i>	5
1.3 对接登录系统	9
1.3.1 登录流程图	9
1.3.2 登陆初始化	9
1.3.3 登陆说明.....	10
1.3.5 用户回话验证.....	10
1.4 提交用户游戏信息.....	11
1.4.1 接口定义.....	11
1.4.2 参数说明.....	11
1.4.3 接口说明.....	11
1.5 对接充值系统.....	12
1.5.1 充值流程图	12
1.5.2 充值接口.....	12
1.6 浮点功能	14
1.7 资源回收接口(游戏退出必须调用).....	15
1.8 充值服务器回调接口开发.....	15
1.8.1 通知充值结果.....	15
1.8.2 签名说明.....	16
1.8.3 合作商响应请求	16
1.8.5 温馨提示.....	17
1.9 运行环境	17
2.0 混淆	17

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：“运营”

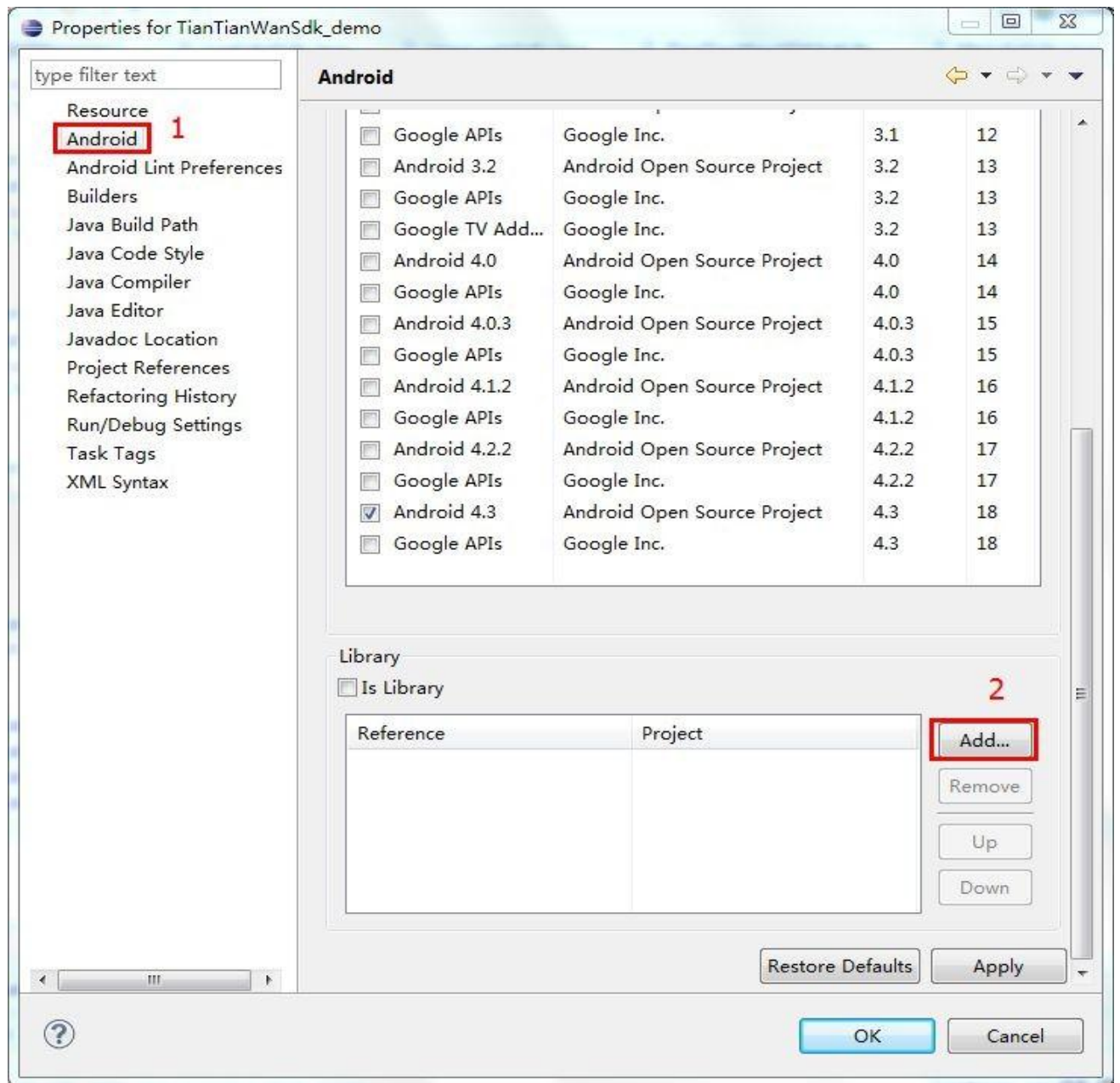
1.2 集成 SDK

集成出 *DM_SdkLib*、拷贝资源文件，配置清单 *AndroidManifest.xml*

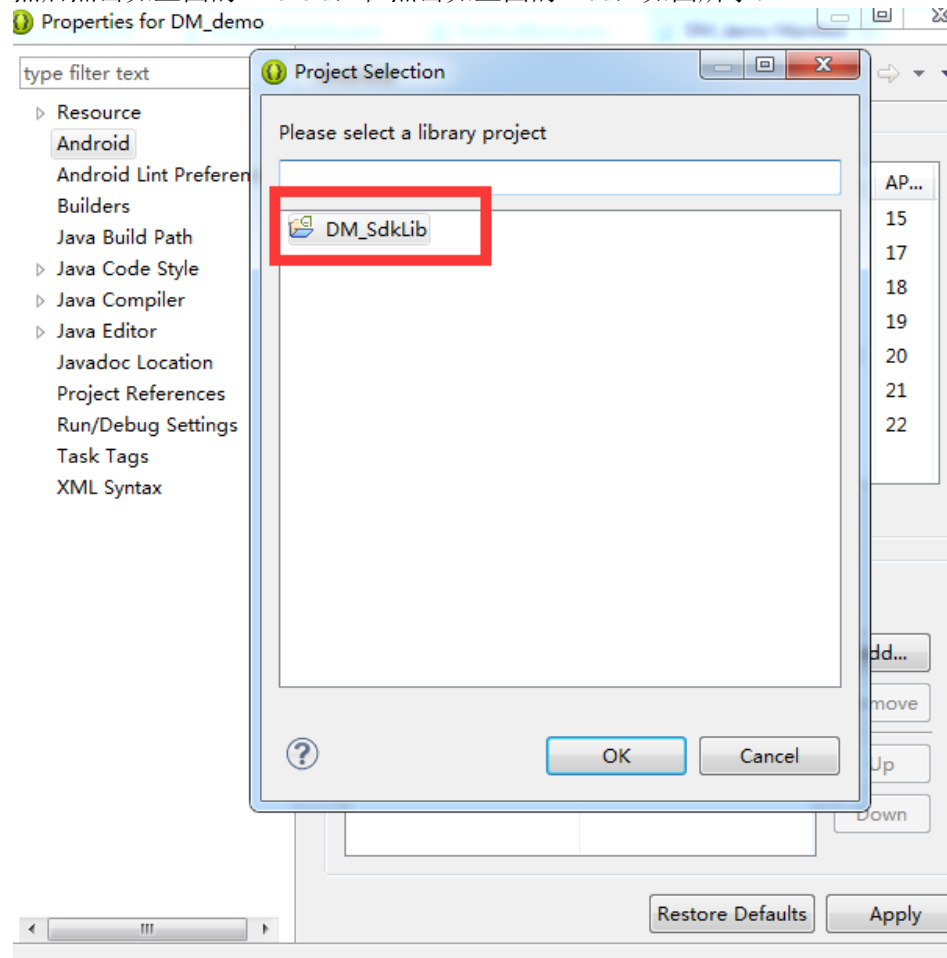
1.2.1 引进DM_SdkLib

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

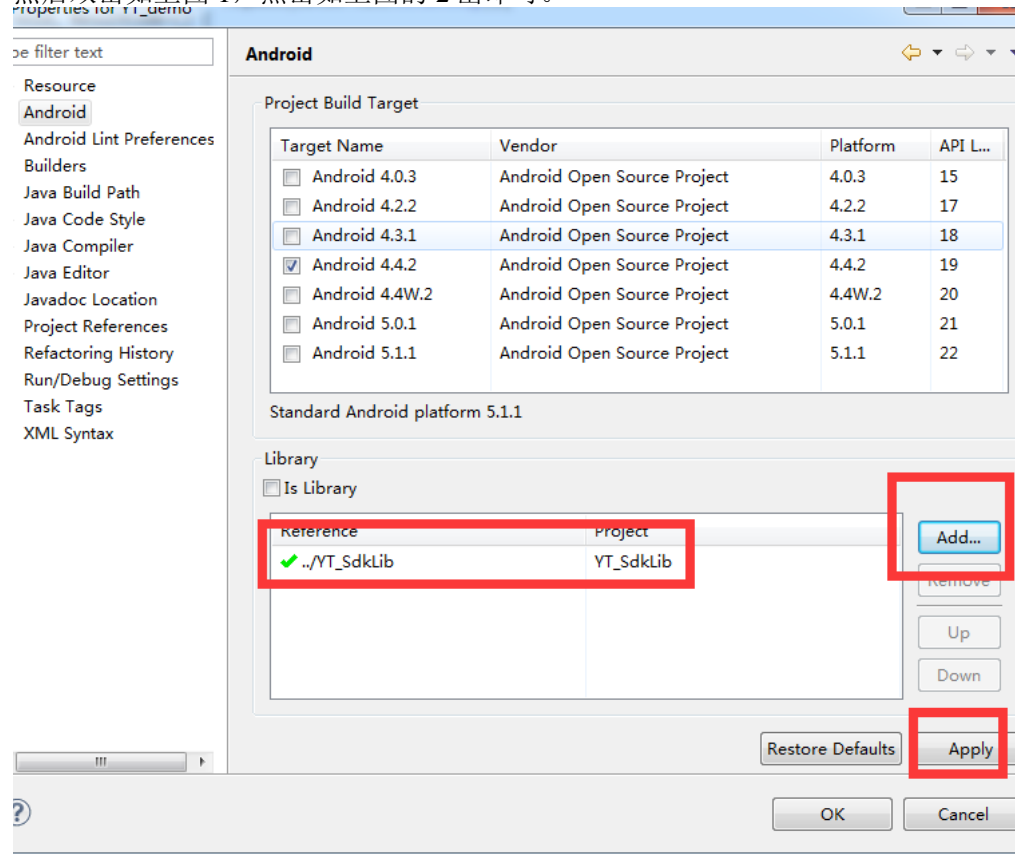
1：右键项目工程，弹出的菜单中 选择最下面的 Properties 弹出页面如图所示：



然后点击如上图的 Android，在点击如上图的 Add，如图所示：

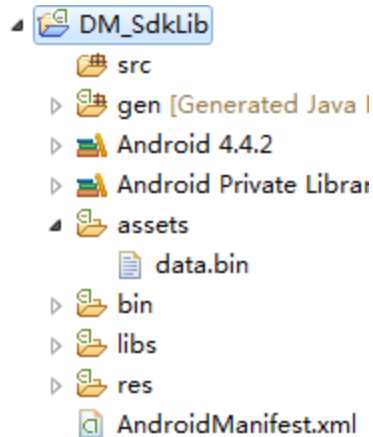


然后双击如上图 1，点击如上图的 2 出即可。



1.2.2 拷贝资源文件

将 *DM_SdkLib* 项目中的 *assets* 文件夹内容拷贝到游戏工程中。



1.2.3 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:
置以下内容:

```
<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"/>
<!-- 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"/>
```

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

```
<!-- 易联支付 -->
    <activity android:name="com.game.sdk.ui.BankEcopayActivity"
        android:theme="@android:style/Theme.NoDisplay"
        android:screenOrientation="portrait"
        android:launchMode="singleTask"
        android:configChanges="keyboardHidden/orientation"
    />
<!-- 支付宝支付 -->
<activity android:name="com.game.sdk.ui.AlipayActivity"
    android:theme="@android:style/Theme.NoDisplay"
    android:screenOrientation="portrait"
    android:launchMode="singleTask"
    android:configChanges="keyboardHidden/orientation"
/>
<!-- 易宝支付 -->
<activity android:name="com.game.sdk.ui.BankpayActivity"
    android:theme="@android:style/Theme.NoDisplay"
    android:screenOrientation="portrait"
    android:launchMode="singleTask"
    android:configChanges="keyboardHidden/orientation"
/>

<activity android:name="com.game.sdk.ui.TCLActivity"
    android:theme="@android:style/Theme.NoDisplay"
    android:screenOrientation="portrait"
    android:launchMode="singleTask"
    android:configChanges="keyboardHidden/orientation"
/>
<activity
    android:name="com.game.sdk.ui.PtbActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.NoTitleBar"
    android:configChanges="keyboardHidden/orientation/screenSize
/keyboardHidden"
/>
<!-- 易宝卡类支付 -->
<activity
    android:name="com.game.sdk.ui.Mobl ieCardActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.NoTitleBar"
    android:configChanges="keyboardHidden/orientation"
/>
    <activity android:name="com.game.sdk.ui.ChargeActivity"
        android:theme="@android:style/Theme.Light.NoTitleBar"
        android:screenOrientation="portrait"
        android:launchMode="singleTask"
        android:configChanges="keyboardHidden/orientation/screenSize
/keyboardHidden"
    />

<activity
    android:name="com.game.sdk.ui.OnkeypayChargeNextActivity"
    android:screenOrientation="portrait"
    android:configChanges="keyboardHidden/orientation"/>

<activity
    android:name="com.game.sdk.ui.Mobl ieCardNextDealActivity"
    android:screenOrientation="portrait"
    android:configChanges="keyboardHidden/orientation"
/>
```

```

<activity android:name="com.game.sdk.ui.LoginActivity"
    android:theme=
        "@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
    android:launchMode="singleTask"
    android:screenOrientation="portrait"
    android:configChanges="keyboardHidden|orientation|screenSize
/keyboardHidden"
/>
<!-- 微信支付 -->
<activity
    android:name="com.game.sdk.ui.NowpayActivity"
    android:theme="@android:style/Theme.NoDisplay"
    android:screenOrientation="portrait"
    android:launchMode="singleTask"
    android:configChanges="keyboardHidden|orientation"/>

<activity
    android:name="com.ipaynow.plugin.activity.PayMethodActivity"
    android:theme="@android:style/Theme.Dialog"
    android:configChanges="orientation|keyboardHidden"
    android:exported="false"
    android:screenOrientation="portrait">
    </activity>

<activity
    android:name="com.ipaynow.plugin.
inner_plugin.prepay_plugin.activity.PrePayActivity"
    android:windowSoftInputMode="adjustResize"
    android:screenOrientation="portrait" >
</activity>

<activity
    android:name="com.ipaynow.plugin.inner_plugin.
prepay_plugin.activity.NotifyActivity"
    android:configChanges="orientation"
    android:exported="false"
    android:screenOrientation="portrait" >
</activity>

<activity
    android:name="com.ipaynow.plugin.
inner_plugin.wechatwp.activity.WeChatNotifyActivity"
    android:theme="@android:style/Theme.NoDisplay">
</activity>

<!-- 浮点 -->
<activity android:name="com.game.sdk.floatwindow.FloatWebActivity"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
    android:configChanges="orientation|keyboardHidden"/>

```

```

<!-- 插件拍照界面 -->
<activity
    android:name="com.payeco.android.plugin.PayecoCameraActivity"
    android:screenOrientation="portrait"/>

<!-- 插件录像界面 -->
<activity
    android:name="com.payeco.android.plugin.PayecoVedioActivity"
    android:screenOrientation="Landscape"
    android:process="com.payeco.android.plugin.vedio"/>

<!-- 浮点 -->
<activity android:name="com.game.sdk.floatwindow.FloatWebActivity"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
    android:configChanges="orientation|keyboardHidden"/>

<!-- 百度插件部分 -->
<service android:name="com.baidu.location.f"
    android:enabled="true"
    android:process=":remote" >
    <intent-filter>
        <action android:name="com.baidu.location.service_v2.2" >
        </action>
    </intent-filter>
</service>

<!-- meta-data 需要写在 application 中 -->
<meta-data android:name="com.baidu.lbsapi.API_KEY"
    android:value="DhNxS47YIr3CNdocGoAcy1ts" />

<!-- 支付宝相关配置,此处按第三方支付添加配置 -->
<activity android:name="com.alipay.sdk.app.H5PayActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind"
    android:windowSoftInputMode="adjustResize|stateHidden" >
    </activity>

<!-- 此处用于配置客户参数信息 -->
<!-- 应用 ID -->
<meta-data android:name="YT_APPID" android:value="1" />

```

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 初始化

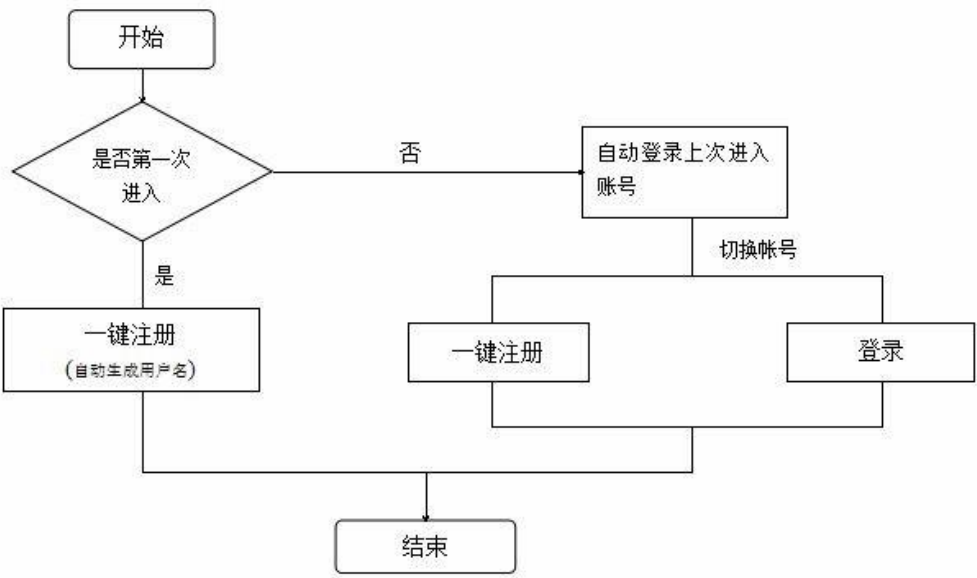
1. 在主线程调用 SDKManager 的 `getInstance(Context ctx)` 静态方法来获取 SDKManager 的实例,进行初始化。

1.3.2 界面初始化

1. 在主线程调用 SDKManager 的实例,实现 `setIsPortrait(int IsPortrait)`进行横竖版界面调整，1 为竖版，2 为横版。

```
sdkmanager.setIsPortrait(1);//1 为竖版，2 为横版
```

1.3.3 登录流程图



1.3.4 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 `public void showLoginView(acontext, boolean isShowQuikLogin, OnLoginListener loginlistener)`方法显示登录界面

参数名	类型	参数说明
<code>acontext</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
<code>onloginListener</code>	<code>OnLoginListener</code>	登录接口的回调，示例如下

```

Sdk.showLoginView(acontext, true, new OnLoginListener() {

    //成功登录回调

    @Override
    public void loginSuccess(LoginCallback logincallback) {

        long logintime=logincallback.logintime;//登录回调时间戳

        String username=logincallback.username;//登录的用户名

        String token=logincallback.token;//口令

    }

    //登录失败回调 可能是用户名密码不正确，也有可能是服务端出现了问题
    @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
token	String	签名口令

1.3.5 用户会话验证

1) 请求地址

正式环境: <http://sdk.93damai.com/cpVerify.php>

2) 调用方式: HTTP POST

3) 接口描述:

验证账号是否为有效的登录用户会话，若有效则返回其账号信息。“游戏客户端”通过“SDK 客户端”获取到参数，传到“游戏服务器”，“游戏服务器”到“SDK 服务器”验证用户会话的有效性，获取用户的账号标识、账号创建者和昵称，供游戏使用。注意：进行接口调用前请确认username是否具备值，如username值为空时请勿调用此接口。

请求内容

字段名称	字段说明	类型	必填	备注
id	请求的唯一标识	int	Y	Unix 时间戳，例：1445502536
appid	游戏 appid	int	Y	
username	用户名	string	Y	
token	签名	string	Y	签名口令

json 请求内容:

```
{
  'id'      : $id,
  'appid'   : $appid,
  'username' : $username,
  'token'   : $token
}
```

返回内容(json 格式)

```
{
  'id' : $verifyid,
  'status': 10,
  'data' => '请求参数错误'
}
```

字段名称	字段说明	类型		备注
id	请求的唯一标识	int		Unix 时间戳，与请求内容中 id 值相同。例：1445502536
status	响应状态	Int		验证状态，请参考下面状态码
data	响应消息	String		返回消息提示

状态码

状态码	说明	错误原因
1	成功	
10	请求参数错误	请求内容格式有误、APPID 有误、签名校验失败等
11	用户未登陆	用户不在线、不存在等

1.4 提交用户游戏信息（必接）

1.4.1 接口定义

`void submitExtendData(String type, JSONObject data)`

示例：

//登录游戏角色成功后调用此段，请根据实际业务数据传入真实数据，以下数据仅是示例

```
try{
    JSONObject jsonExData = new JSONObject();
    jsonExData.put("service", "游戏一区-逍遥谷");
    jsonExData.put("role", "令狐冲");
    jsonExData.put("grade", "99");

    SDK.getInstance(this).submitExtendData(1, jsonExData);
} catch (Exception e) {
    //处理异常
}
```

1.4.2 参数说明

参数	说明
type	扩展数据的种类
data	扩展数据的内容，不同各类的扩展数据具有不同的数据内容定义，具体见下表， 请根据实际业务数据传入真实数据。

扩展参数定义

数据种类 (type)	数据内容定义 (data)	说明
1	{ "service":"string", //游戏区服，必填 "role":"string", //游戏角色，必填 "grade":"string", //游戏等级，必填 }	游戏中为用户成功登录游戏角色时，提交该“游戏登录角色”数据

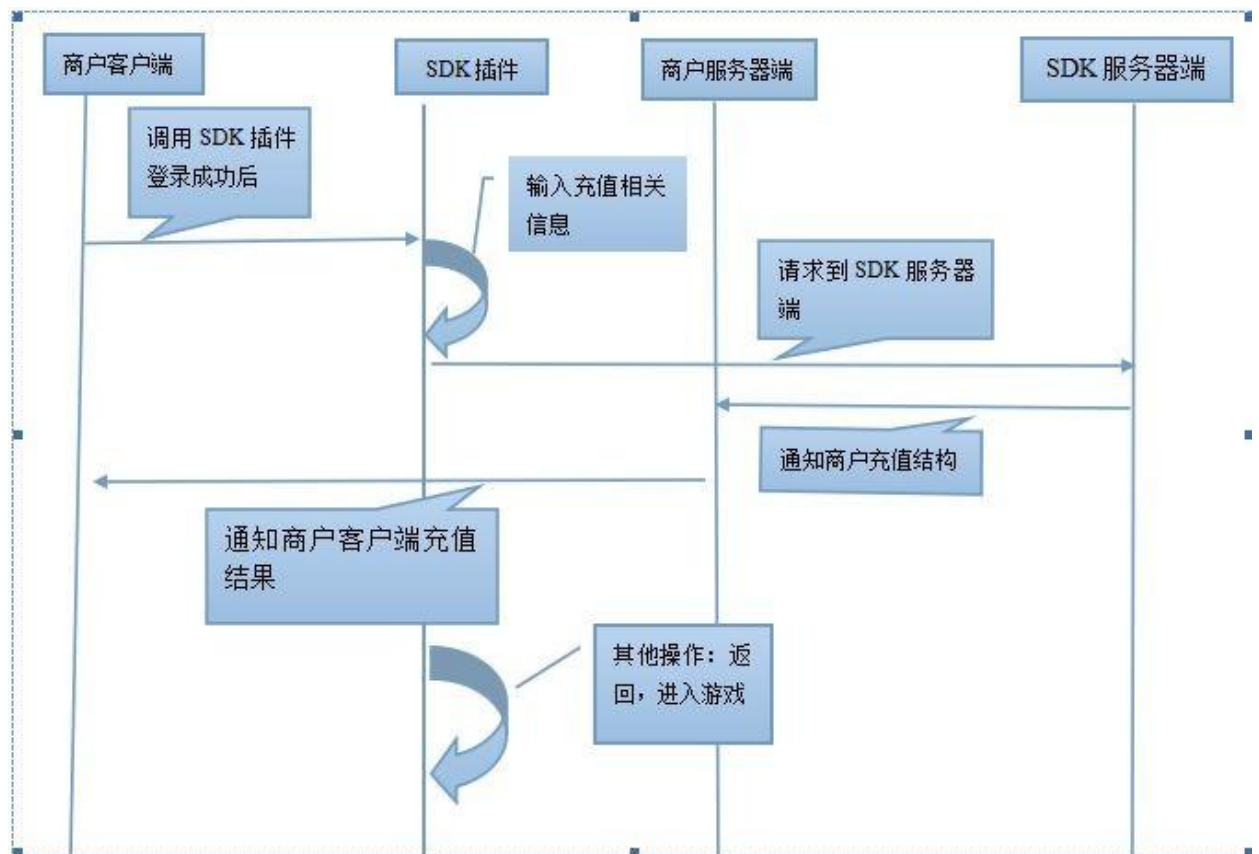
注：如没有区服，则数据类型为0，区服角色为0，比如棋牌类游戏。

1.4.3 接口说明

CP 在成功分配用户角色信息后，调用此接口，提交用户游戏相关信息。

1.5 对接充值系统

1.5.1 充值流程图



1.5.2 充值接口

1. 调用 SDKManager 的实例方法来获取。
2. 调用 SDKManager 的 `public void showPaymentView(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(传入整数字符, 如 1, 2)
serverid	String	是	合作商服务器 id
productname	String	是	产品名字
productdesc	String	是	产品描述
paymentListener	OnPaymentListener	是	对充值是否成功进行监听

充值示例如下：

```
sdk.showPaymentView(ctx , roleid,
                    money,
                    serverid,
                    productname,
                    productdesc,
                    new
                    OnPaymentListener() {

@Override
public void paymentSuccess(PaymentCallbackInfo paymentCallbackInfo)
{
    int money=paymentCallbackInfo.money;// 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

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

@Override
protected void onResume() {
    // TODO Auto-generated method stub
    sdkmanager.showFloatView();//显示悬浮窗口
    super.onResume();
}
```

1.7 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)

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 我方，由我方发起请求，合作商响应请求。请求协议采用 **http post**

1.8.1 通知充值结果

玩家在游戏中使用手游 SDK 进行充值，SDK 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](#)），SDK 服务 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 **urlencode** 处理，urlencode 编码统一为 **UTF-8**，注意区分大小写，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	订单号	1
username	String	30	登录账号	2
appid	int	11	游戏 ID	3
roleid	String	30	游戏角色 ID	4
serverid	int	11	服务器 ID	5
amount	int	11	成功充值金额，单位(元)	6
paytime	int	11	玩家充值时间，时间戳形式，如1394087000	7
attach	String		商户拓展参数	8
productname	String		商品描述	9
sign	String	32	参数签名（用于验签对比）	10

1.8.2 签名说明

合作商接收到 SDK 方 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&appid=6&roleid=zhangsanfeng&serverid=1&amount=1&paytime=20130101125612&attach=test&productname=test&appkey=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 SDK 方 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 SDK 方发出的请求后，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"，SDK 方 会启用系统重发机制](#)

1.7.4 温馨提示

1. 由于 我方 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，我方 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我方同步的 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 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/4G/WIFI 等等

2.0 混淆

SDK 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 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: `-libraryjars libs/*.jar` 以避免 SDK 的相关的 jar 包被混淆。