

摩格科技SDK开发手册v2.1

- [摩格科技SDK开发手册v2.1](#)
 - [SDK开发包目录清单](#)
 - [1. 编写目的](#)
 - [2. 集成SDK](#)
 - [3. 摩格SDK初始化](#)
 - [4. 对接登录系统](#)
 - [5. 对接充值系统](#)
 - [6. 充值服务端回调接口](#)
 - [7. 混淆说明](#)

SDK开发包目录清单

```
摩格科技SDK开发手册v2.1.pdf
图标角标.png
更新文档.txt
sdk/
  AndroidManifest.xml
  assets/
    MogePay.apk
  libs/
    armeabi/
      libytsdk.so
    mgsdk-2016****.jar
    libammsdk.jar
    alipaySdk-20151112.jar
    android-support-v4.jar
  res/
    drawable/ **
    drawable-hdpi/ **
    layout/ **
    values/ **
  sdk-demo/
    /* 范例项目 */

MogeDemo.apk
```

1. 编写目的

本文档提供给游戏合作商对接的同学使用，文档主要分为以下7个部分：

1. 集成SDK；
2. 摩格SDK初始化；
3. 对接登录系统；
4. 对接充值系统；
5. 充值服务端回调接口；
6. 混淆说明

2. 集成SDK

• 2.1 步骤一：导入libs

将 sdk/libs/目录中的jar文件以及so文件复制到项目工程的libs对应目录中；

• 2.2 步骤二：复制资源文件

复制 sdk/res/里各个子目录中的文件至项目工程的res对应各个文件夹里；

复制 **sdk/assets/** 里的MogePay.apk文件至项目工程的assets文件夹里；

• 2.3 步骤三：配置清单AndroidManifest.xml

在工程AndroidManifest.xml设置以下权限清单：（如有误差，请以DEMO为准）

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

```

在工程AndroidManifest.xml的标签段内配置以下内容：（如有误差，请以DEMO为准）

```

<activity
    android:name="com.game.sdk.ui.ChargeActivity"
    android:configChanges="keyboardHidden|orientation"
    android:launchMode="singleTask"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Light.NoTitleBar" />
<activity
    android:name="com.game.sdk.ui.OnkeypayChargeNextActivity"
    android:configChanges="keyboardHidden|orientation"
    android:screenOrientation="behind" />
<activity
    android:name="com.game.sdk.ui.MoblieCardNextDealActivity"
    android:configChanges="keyboardHidden|orientation"
    android:screenOrientation="behind" />
<activity
    android:name="com.game.sdk.ui.LoginActivity"
    android:configChanges="keyboardHidden|orientation"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
    />
<!-- 浮点 -->
<activity
    android:name="com.game.sdk.floatwindow.FloatWebActivity"
    android:configChanges="orientation|keyboardHidden"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
    />
<activity
    android:name="com.alipay.sdk.auth.AuthActivity"
    android:configChanges="orientation|keyboardHidden|navigation"
    android:exported="false"
    android:screenOrientation="behind" >
</activity>

<!-- alipay sdk begin -->

```



```

<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"
    android:launchMode="singleTask"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.NoDisplay" />

<!-- alipay sdk end -->
<activity
    android:name="com.game.sdk.ui.PtbActivity"
    android:configChanges="keyboardHidden|orientation"
    android:screenOrientation="behind"
    android:theme="@android:style/Theme.NoTitleBar" />

<service android:name="com.game.sdk.YTAppService" >
    <intent-filter android:priority="1000" >
        </intent-filter>
    </service>
<service android:name="com.game.sdk.xutils.download.DownloadService" >
    <intent-filter>
        <action android:name="download.service.action" />
    </intent-filter>
</service>
<activity
    android:name="com.game.sdk.ui.PayWXProxyActivity"
    android:launchMode="singleTask"
    android:exported="true"
    android:theme="@style/mgsdk_style_transparent"
    android:screenOrientation="portrait"/>

<!-- meta-data需要写在application中 -->
<meta-data
    android:name="MG_APPID"
    android:value="1" />
<meta-data
    android:name="MG_GAMEID"
    android:value="1" />
<meta-data
    android:name="MG_AGENT"
    android:value="defalut" />
<meta-data
    android:name="MG_LOGOUT_VISIBLE"
    android:value="true" />

```

增加一个meta-data: **MG_LOGOUT_VISIBLE**, 用于配置浮标中退出按钮是否显示

在工程AndroidManifest.xml的标签段内配置以下内容: (如有误差, 请以DEMO为准)

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

注意: 接入SDK项目的**Android SDK Target**请控制在 8 - 20 之间; 即 Android 2.2 以上、5.0 以下。

3. 摩格SDK初始化

• 3.1 初始化

合作商需在onCreate中调用MGSDKManager的getInstance(Context)静态方法来获取摩格SDK的实例 (MGSDKManager.getInstance(ctx)可以多次调用, 不会影响程序运行性能), 并调用SDKManager实例的初始化方法init(Context), 来完成SDK的初始化工作; 示例如下:

```
MGSDKManager sdkmanager;
sdkmanager = MGSDKManager.getInstance(this);
sdkmanager.init(this);
sdkmanager.onCreate();
```

• 3.2 生命周期

请在Activity对应的生命周期中加上如下方法: (Demo中可查看所有生命周期相关方法)

```
sdkmanager.onCreate();

sdkmanager.onConfigurationChanged(Configuration newConfig);

sdkmanager.onNewIntent(Intent intent);

sdkmanager.onActivityResult(int requestCode, int resultCode, Intent data);
```

```
sdkmanager.onPause();

sdkmanager.onDestroy();

sdkmanager.onStop();

sdkmanager.onStart();

sdkmanager.onResume();

sdkmanager.onRestart();

sdkmanager.onBackPressed();
```

4. 对接登录系统

• 4.1 设置SDK注销回调

在SDK初始化结束之后，应先监听注销事件的回调：

注意：该监听器仅监听浮标的退出回调，并不监听 `sdkmanager.logout()`；

```
sdkmanager.setUserLogoutListener(new OnUserLogoutListener() {

    @Override
    public void onLogoutSuccess(int code, String msg) {
        // 注销成功
        Toast.makeText(getApplication(), "注销成功: " + msg, Toast.LENGTH_LONG).show();
    }

    @Override
    public void onLogoutFailure(int code, String msg) {
        // 注销失败
    }

});
```

• 4.2 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MGSDKManager.getInstance(context).login()方法显示登录界面，参数详解：

参数名	类型	参数说明
context	Context	上下文context实例
isShowQuickLogin	boolean	是否显示快捷登录 (true:显示)
loginListener	OnLoginListener	登录接口的回调

示例如下：

```
sdkmanager.login(MainActivity.this, true, new OnLoginListener() {  
  
    @Override  
    public void onLoginSuccess(LoginResult result) {  
        String sign = result.sign;// 登录成功返回的签名,需要做验证处理  
        long loginTime = result.logintime;// 登录回调时间戳  
        String username = result.username;// KD 登录的用户名  
        String memkey = result.memkey;// 平台登录返回的Token KEY, 用于登  
录验证  
        Toast.makeText(getApplication(), "登录成功: " + result.username, Toast.LENGTH_LONG).show();  
    }  
  
    @Override  
    public void onLoginFailure(int code, String msg) {  
        Logger.msg("onLoginFailure");  
    }  
});
```

2. 登录失败的信息解析

参数名	类型	参数说明
code	int	登录失败错误码
msg	String	登录失败的消息提示

登录错误码列表：

CODE	说明
-1	帐号为空
-2	密码错误

-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	未知错误

3. 登录成功的信息体LoginResult解析

参数名	类型	参数说明
username	String	登录成功后，用户的用户名（唯一）
logintime	long	用户登录的时间戳
sign	String	签名，用于做登录验签比对
memkey	String	平台登录返回的Token KEY，用于登录验证

4. 登录参数签名规则

参数名	类型	参数说明	签名顺序
username	String	用户帐号	1
appkey	String	应用的appkey	2
logintime	String	登录时间戳	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：

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

• 4.3 退出登录(注销帐号)

主动注销帐号直接调用MGSDKManager的 `public void logout()` 方法即可；

```
sdkmanager.logout();
```


• 4.4 SDK服务端登录二次验证

使用SDK登录功能后，合作商可以根据需求，利用SDK登录返回的信息体进行二次登录验证；

API地址：<http://game.6y.com.cn/sdk/islogin.php>

参数如下(POST):

参数名	参数类型	说明
username	String	用户名
memkey	String	平台登录返回的Token KEY

验证返回结果说明(String):

返回值	说明
success	验证登录成功，该用户已经登录
fail	验证失败，该用户未登录

5. 对接充值系统

• 5.1 充值接口说明

1. 调用MGSDKManager的

```
public void pay(context, roleid, money, serverid, productName, productDesc, attach, paymentListener)
```

方法显示充值界面；参数说明：

参数名	类型	是否必须	参数说明
context	Context	YES	上下文context实例
roleid	String	YES	角色id
money	String	YES	充值金额(整数不可含小数点)
serverid	String	YES	合作商服务器id
productName	String	YES	产品名字
productDesc	String	YES	产品描述

attach	String	YES	合作商扩展参数(透传参数)
listener	OnPaymentListener	YES	充值结果监听器

充值接口使用示例代码:

```
sdkmanager.pay(MainActivity.this, "roleid", "1", "serverid", "一堆钻石", "一堆就是一堆", "商户透传参数", new OnPaymentListener() {

    @Override
    public void onPaySuccess(PaymentResult result) {
        Logger.msg("onPaySuccess");
        Toast.makeText(getApplication(), "充值金额数: " + result.money + " 消息提示: " + result.msg, Toast.LENGTH_LONG).show();
    }

    @Override
    public void onPayFailure(int code, String msg, String money) {
        Logger.msg("onPayFailure");
        Toast.makeText(getApplication(), "错误码:" + code + " 错误码:" + msg + " 金额: " + money, Toast.LENGTH_LONG).show();
    }
});
```

2. 充值失败信息解析

参数名	类型	参数说明
code	String	状态码
money	int	意图充值金额
msg	String	充值失败消息提示

3. 充值成功信息体PaymentResult解析

参数名	类型	参数说明
money	int	充值金额数
msg	String	充值结果描述

6. 充值服务端回调接口

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给摩格SDK，由摩格SDK发起请求，合作商响应请求；请求协议采用HTTP POST；

• 6.1 通知充值结果

玩家在游戏中使用摩格SDK进行充值，摩格SDK在处理完充值流程之后，主动请求通知游戏方，游戏方若未返回接收成功（接收成功判断请参考5.3合作商响应请求），摩格SDK会启动系统重发机制(一共发送五次，如果五次均没有响应，将由工作人员补单)。注意只有充值成才会计回调通知结果，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登录帐号	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(详看下方参数说明)	6
amount	int	11	成功充值金额(单位：元)	7
paytime	int	11	玩家充值时间戳	8
attach	String		商户扩展参数	9
appkey	String	32	应用的appkey	10
sign	String	32	参数签名（用于签名比对）	-

• 6.2 SIGN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=14574180753510246&username=d545218253&gameid=12&roleid=superSdkDemo%E6%B5%8B%E8%AF%95&serverid=1&paytype=ptb&amount=1&paytime=1457418075&attach=%7B%22appid%22%3A%2210281001%22%2C%22channelkey%22%3A%22mgsdk%22%2C%22orderid%22%3A%2220160308142011121060%22%7D&appkey=91bac46a9b70bd2db563cc483d443ba3")
```

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 摩格SDK 提供，登录、充值使用同一个 *appkey*

• 6.3 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：

```
echo("success");
```

代码	代码描述
success	接收并验证成功
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*”，将会启用系统重发机制

• 6.4 支付类型参数表

支付代码	支付类型描述
ylzf	易联支付
zfb	支付宝快捷支付
ptb	平台币支付

• 6.5 温馨提示

- 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；

- 。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。

7. 混淆说明

摩格 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 XXSDK 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 XXSDK 的相关的 jar 包被混淆。