

梦工厂 APK 游戏支付 SDK 对接文档(CP)

V5.0

版本号	时间	变更内容
5.0	2021-07-18	创建

梦工厂 APK 游戏支付 SDK 对接文档(CP).....	1
集成流程.....	3
工程设置.....	3
API.....	3
SDK 初始化.....	4
登录.....	4
设置游戏角色信息.....	5
支付.....	6

集成流程

1. CP 在梦工厂后台(<http://mini.mgc-games.com>)登录, 在“我的游戏/应用”中点击“新建游戏”, 可以注册游戏信息, 如果需要支付, 请填写支付回调地址. 游戏新建后, 在“我的游戏/应用”界面展开游戏, 可以看到“APPID”
2. 集成沙盒 sdk 到您的游戏
3. sdk 的 zip 解压后, testenv 目录下的 sandbox.apk 是一个沙箱测试环境, 首先通过“adb install sandbox.apk”安装到您的手机
4. 将您游戏打包后的 apk 安装到手机, 或者拷贝到 sd 卡根目录(adb push xxx.apk /sdcard)
5. 运行沙箱测试环境, 首界面有两个按钮“从已安装 App 中克隆安装”, “从 SDCard 安装 APK 文件”. 如果您使用 SD 卡请点击后者, 如果不是点击前者. 找到自己的应用或者 apk, 点击旁边的安装, 完成后. 返回首界面即能看到您的应用, 点击即可运行.
6. 如果您更新了应用, 请在 sandbox 中先删除原来的, 再执行第 5 步重新安装

工程设置

1. 将 sdk 的 zip 解压后, libs 目录下的 leto.sandbox.sdk-x.y.z.aar 拷贝到您工程的 libs 目录下并添加依赖(参考代码, 请根据您的具体写法修改):

```
dependencies {  
    implementation fileTree(include: ['*.aar'], dir: 'libs')  
}
```

2. 将游戏的 APPID 添加到您应用的 AndroidManifest.xml 的<application>下中, 如下:

```
<meta-data android:name="MGC_SANDBOX_GAME_ID" android:value="游戏的 APPID"/>
```

API

SDK 初始化

在您应用的主 Activity 中(注意是主 Activity, 即处理 android.intent.action.MAIN 的 Activity), 在 onCreate 方法

里添加以下代码进行初始化:

```
if(LSBSdk.init(this)) {  
    // 当前处于沙箱环境  
} else {  
    // 当前处于独立环境  
}
```

init 方法会返回 boolean, true 表示当前处于沙箱环境中, false 表示当前不在沙箱中。不在沙箱中时, 大部分

api 调用不会有效果。

登录

调用方式:

```
// 添加登录回调  
LSBSdk.addLSBSignInListener(this);  
  
// 开始登录  
LSBSdk.startLSBSignIn()
```

ILSBSignInListener 接口说明:

方法	说明
<code>void onLSBSignInSuccess (LSBUserInfo userInfo);</code>	登录成功后调用 userInfo: 用户信息类, 见下表

方法	说明
<code>void onLSBSignInFailed(String errMsg);</code>	登录失败后调用 errMsg: 错误信息
<code>void onLSBSignInCancelled();</code>	登录取消时调用

LSBUserInfo 说明:

字段	说明
<code>String userToken</code>	令牌
<code>String memId</code>	用户 id

设置游戏角色信息

在登录完成后, 您应该在成功回调里设置游戏的角色信息, 在游戏角色信息发生变化时也需要设置. 角色

信息在使用支付时需要使用. 角色信息的所有字段都是可选字段, 不提供则为空字符串.

调用方式:

```
LSBRoleInfo roleInfo = new LSBRoleInfo(); // 创建 roleinfo
```

```
roleInfo.setServerID("1");// 服务器 ID, 其值必须为数字字符串
```

```
roleInfo.setServerName("火星服务器");// 服务器名称
```

```
roleInfo.setGameRoleName("裁决之剑");// 角色名称
```

```
roleInfo.setGameRoleID("1121121");// 角色 ID
```

```
roleInfo.setGameUserLevel("12");// 等级
```

```
roleInfo.setVipLevel("Vip4");// VIP 等级
```

/

```
roleInfo.setGameBalance("5000");// 角色现有金额
```

```
roleInfo.setPartyName(""); // 公会名字
```

```
roleInfo.setRoleCreateTime(String.valueOf(System.currentTimeMillis()));
```

```
LSBSdk.setLSBRoleInfo(roleInfo); // 调用 LSBSdk 的 api
```

LSBRoleInfo 说明:

字段	说明
int roleType	1 为进入游戏, 2 为创建角色, 3 为角色升级, 4 为退出, 5 其他
String serverId	游戏服务器 id, 默认为 0
String serverName	所在服务器名称
String roleId	角色 id
String roleName	角色名称
String partyName	工会、帮派名称, 如果没有, 请填空字符串""
String roleLevel	角色等级, 如果没有, 请填 0
String roleVip	vip 等级, 如果没有, 请填 0。
String roleBalance	用户游戏币余额, 如果没有账户余额, 请填 0。
String roleLevelCreateTime	创建角色的时间戳
String roleLevelModifyTime	角色等级变化时间戳

支付

调用方式:

```
// 添加支付事件回调
```

```
LSBSdk.addLSBPayListener(this);
```

```
// 开始支付
```

```
LSBOrderInfo info= new LSBOrderInfo(); // 创建 order info
```

```
info.setCpOrderId("20161028111"); // cp 自定义 order id
```

```
info.setProductPrice(0.01f); // 单位元, 可以精确到分
```

```
info.setProductCount(1);
```

```
info.setProductId("1");
```

```
info.setProductName("元宝");
```

```
info.setProductDesc("很好");
```

```
info.setExchangeRate(1);
```

```
info.setCurrencyName("金币");
```

```
info.setExt("穿透参数, 会传递给成功回调");
```

```
LSBSdk.startLSBPay(info); // 调用 startLSBPay
```

LSBOrderInfo 说明:

字段	说明
String productId	商品 id
String productName	商品名称
String productDesc	商品描述
String cpOrderId	订单 id, cp 负责生成
int productCount	购买数量

字段	说明
float productPrice	商品价格，以元为单位;建议传入整数,可以保留两位小数
String ext	额外参数，会在支付成功后传递给成功回调
String currencyName	游戏货币名称，默认为金币
int exchangeRate	游戏币与人民币兑换比率,例如 100 为 1 元人民币兑换 100 游戏币。默认为 1

获取渠道 id

如果某些场合需要梦工厂的渠道 id, 可以通过以下 api 获取:

```
String channelId = LSBSdk.getChannelId();
```