

快发助手SDK接入文档

版本号	修改说明	作者	时间
v3.0	文档创建	HooRang	2015-12-07

名词解释

名词	解释
母包	母包 = 纯净的游戏包 + 快发sdk，也就是游戏接好快发sdk之后就是一个母包
打包工具	快发提供的一个win/mac os打包客户端，渠道包由该工具进行生成
签名文件	生成apk时需要使用的*.keystore文件

前言

业务功能、统计、生命周期函数模块如非特别标注“非必接”均表示默认为必接接口，推送、分享插件模块按照cp意愿，自愿选择是否接入；

引入sdk

以library的方式引入sdk库项目(lib_Platform_SDK_v版本号)

Unity的项目请注意：lib_Platform_SDK_v版本号.jar中包含有assets下面的资源，由于Unity项目打包时无法自动把这里的资源拷贝到项目中，所以需要把jar解压出来手动拷贝到Assets-Plugins-Android-assets目录中

配置AndroidManifest.xml

一、权限配置

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
<uses-permission android:name="android.permission.READ_LOGS"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

二、Application配置

```
SDK默认Application为:com.hjr.sdkkit.framework.mw.app.SDKBaseApplication;
如有自定义的Application:需要继承
com.hjr.sdkkit.framework.mw.app.SDKBaseApplication;
将默认的或者是自定义的Application配置到AndroidManifest.xml中Applicataion节点的
android:name当中;
```

三、修改HJR_GAMEKEY

```
<meta-data android:name="HJR_GAMEKEY" android:value="填入申请的gamekey" />
```

客户端接口接入

业务功能接口

一、初始化

通常在游戏第一个Activity的onCreate方法中调用

```
HJRSDKKitPlatformCore sdkObj =
HJRSDKKitPlateformCore.initHJRPlateform(Activity
activity,HJRSDKKitPlateformCallBack callBack);
```

参数	参数说明
activity	游戏当前activity对象
callBack	初始化全局回调对象，所有sdk接口均通过该对象中的接口方法向外进行回调

二、登录

在初始化成功之后方可以调用；这里请注意， 点击设备物理返回键可以取消登录并且sdk在这
种情况下不会给出任何回调， 并且注意处理连续点击按钮情况， 可以参见demo

```
sdkObj.User.login(Activity activity);
```

参数	参数说明
activity	游戏当前activity对象

三、支付

在购买商品时调用，请注意处理连续点击情况，可以参见demo

```
ParamsContainer pc = new ParamsContainer();
pc.putInt(ParamsKey.KEY_PAY_AMOUNT, );
pc.putInt(ParamsKey.KEY_PAY_PRODUCT_NUM, );
pc.putString(ParamsKey.KEY_PAY_ORDER_ID, "");
pc.putInt(ParamsKey.KEY_PAY_PRODUCT_ID, );
pc.putString(ParamsKey.KEY_PAY_PRODUCT_NAME, "");
pc.putString(ParamsKey.KEY_PRODUCT_DESC, "");
pc.putString(ParamsKey.KEY_EXTINFO, "");
sdkObj.Pay.pay(pc);
```

参数	参数说明
ParamsKey.KEY_PAY_AMOUNT	支付金额，整形，以元为单位
ParamsKey.KEY_PAY_PRODUCT_NUM	购买数量，整形，通常为1
ParamsKey.KEY_PAY_ORDER_ID	游戏方自定义订单号，字符串，会通过服务器回传给cp服务器
ParamsKey.KEY_PAY_PRODUCT_ID	商品id，整形，多个充值档位不允许重复
ParamsKey.KEY_PAY_PRODUCT_NAME	当前购买商品的商品名称，字符串
ParamsKey.KEY_PRODUCT_DESC	当前购买商品的商品描述，字符串
ParamsKey.KEY_EXTINFO	扩展参数，字符串，透传给cp服务器，可以为“”

四、注销

需要在游戏中添加按钮，点击后来调用

```
sdkObj.User.logout();
```

五、用户中心

需要在游戏中添加按钮，点击后来调用

```
sdkObj.User.userCenter();
```

六、退出游戏

在点击设备物理返回键中调用，在这里只能保留sdk的一个退出确认提示框

```
sdkObj.Base.exitGame(Activity activity);
```

参数	参数说明
activity	游戏当前activity对象

七、重写Activity生命周期函数

以下生命周期函数原封不动的拷贝到游戏的主Activity当中

```
@Override
protected void onResume() {
    super.onResume();
    if (sdkObj != null) {
        sdkObj.LifeCycle.onResume();
    }
}

@Override
protected void onPause() {
    super.onPause();
    if (sdkObj != null) {
        sdkObj.LifeCycle.onPause();
    }
}

@Override
protected void onStop() {
    super.onStop();
    if (sdkObj != null) {
        sdkObj.LifeCycle.onStop();
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    if (sdkObj != null) {
        sdkObj.LifeCycle.onDestroy();
    }
}

}
```

```

@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    if (sdkObj != null) {
        sdkObj.LifeCycle.onConfigurationChanged(newConfig);
    }
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    if (sdkObj != null) {
        sdkObj.LifeCycle.onSaveInstanceState(outState);
    }
}

@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    if (sdkObj != null) {
        sdkObj.LifeCycle.onNewIntent(intent);
    }
}

```

八、全局回调对象HJRSDKKitPlatformCallBack

- initCallBack(int retStatus, String retMessage)

调用HJRSDKKitPlatformCore.initHJRPlatform(Activity activity ,PlatformSDKCallBack callBack)后回调

参数	参数说明	是否可能为空
retStatus	接口调用返回状态 初始化成功： HJRSDKKitPlatformCallBack.STATUS_SUCCESS 初始化失败： HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是

- loginCallBack(String loginUserId,String loginUserName, String loginAuthToken, String loginOpenId, boolean switchUserFlag , int retStatus, String retMessage)

调用sdkObj.User.login()后回调		
参数	参数说明	是否可能为空
loginUserId	某个渠道用户唯一标识	否
loginUserName	某个渠道用户名	是
loginAuthToken	用户验证token，可以用该值向sdk服务端验证用户有效性	否
loginOpenId	全平台用户唯一标识	否
switchUserFlag	true:来自切换账号的登录；false:正常登录	否
retStatus	接口调用返回状态 登录成功： HJRSDKKitPlatformCallBack.STATUS_SUCCESS 登录失败： HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是

- payCallBack(String payKitOrderId, int retStatus,String retMessage)

调用sdkObj.Pay.pay()后回调		
参数	参数说明	是否可能为空
payKitOrderId	平台支付订单号	否
retStatus	接口调用返回状态 支付成功： HJRSDKKitPlatformCallBack.STATUS_SUCCESS 支付失败：HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是

- logoutCallBack(int retStatus, String retMessage)

调用sdkObj.User.logout()后回调		
参数	参数说明	是否可能为空
retStatus	接口调用返回状态 注销成功： HJRSDKKitPlatformCallBack.STATUS_SUCCESS 注销失败：HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是

- exitGameCallBack(int retStatus, String retMessage)

调用sdkObj.Base.exitGame()后回调		
参数	参数说明	是否可能为空
retStatus	接口调用返回状态 点击确定: HJRSDKKitPlatformCallBack.STATUS_SUCCESS 点击取消: HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是

- getOrderResultCallBack(String orderStatus,int retStatus, String retMessage)

调用sdkObj.Pay.getOrderInfo(pc)的回调		
参数	参数说明	是否可能为空
retStatus	接口调用返回状态 点击确定: HJRSDKKitPlatformCallBack.STATUS_SUCCESS 点击取消: HJRSDKKitPlatformCallBack.STATUS_FAIL	否
retMessage	接口调用返回说明信息	是
orderStatus	订单状态: 0: 整个充值流程已经成功走完;4: 标识充值成功但是添加道具失败; 否则表示失败	否

统计接口

一、进入游戏

在每次进入游戏时调用

```
ParamsContainer pc = new ParamsContainer();
pc.putString(ParamsKey.KEY_ROLE_ID, "");
pc.putString(ParamsKey.KEY_ROLE_NAME, "");
pc.putInt(ParamsKey.KEY_ROLE_LEVEL, );
pc.putString(ParamsKey.KEY_SERVER_ID, "");
pc.putString(ParamsKey.KEY_SERVER_NAME, "");
sdkObj.Collections.onDatas(DataTypes.DATA_ENTER_GAME, pc);
```

参数	参数说明	是否可以为空
ParamsKey.KEY_ROLE_ID	当前登录用户角色id	否
ParamsKey.KEY_ROLE_NAME	当前登录用户角色名	否
ParamsKey.KEY_ROLE_LEVEL	当前登录用户角色等级	否
ParamsKey.KEY_SERVER_ID	当前登录区服id	否
ParamsKey.KEY_SERVER_NAME	当前登录区服名	否

二、创建角色

在游戏中完成创建角色操作之后调用

```
ParamsContainer pc = new ParamsContainer();
pc.putString(ParamsKey.KEY_ROLE_ID, "");
pc.putString(ParamsKey.KEY_ROLE_NAME, "");
pc.putString(ParamsKey.KEY_SERVER_ID, "");
pc.putString(ParamsKey.KEY_SERVER_NAME, "");
sdkObj.Collections.onDatas(DataTypes.DATA_CREATE_ROLE, pc);
```

参数	参数说明	是否可以为空
ParamsKey.KEY_ROLE_ID	创建的角色id	否
ParamsKey.KEY_ROLE_NAME	创建的角色名	否
ParamsKey.KEY_SERVER_ID	当前登录用户所在区服id	否
ParamsKey.KEY_SERVER_NAME	当前登录用户所在区服名	否

三、角色升级

在游戏中角色完成升级时调用，需要实时调用；

```
ParamsContainer pc = new ParamsContainer();
pc.putString(ParamsKey.KEY_ROLE_LEVEL, "");
sdkObj.Collections.onDatas(DataTypes.DATA_ROLE_UPGRADE, pc);
```

参数	参数说明	是否可以为空
ParamsKey.KEY_ROLE_LEVEL	角色升级后的等级	否

四、支付统计

在客户端收到支付成功回调之后调用

```
ParamsContainer pc = new ParamsContainer();
pc.putInt(ParamsKey.KEY_AMOUNT, );
pc.putString(ParamsKey.KEY_ORDERNUMBER, "");
pc.putString(ParamsKey.KEY_PRODUCT_DESC, "");
sdkObj.Collections.onDatas(DataTypes.DATA_PAY, pc);
```

参数	参数说明	是否可以 为空
ParamsKey.KEY_AMOUNT	支付金额，以实际付款金额为准	否
ParamsKey.KEY_ORDERNUMBER	订单号，可以使用平台支付返回订单号/游戏自己的订单号	否
ParamsKey.KEY_PRODUCT_DESC	支付订单描述	是

分享模块

一、打开分享

```
ParamsContainer pc = new ParamsContainer();
pc.putString(ParamsKey.KEY_SHARE_TITLE, "");
pc.putString(ParamsKey.KEY_SHARE_DESCRIPTION,"");
pc.putString(ParamsKey.KEY_SHARE_TEXT, "");
pc.putString(ParamsKey.KEY_SHARE_TARGETURL, "");
pc.putString(ParamsKey.KEY_SHARE_IMAGEURL, "");
sdkObj.Share.share(pc);
```

参数	参数说明	是否可以 为空
ParamsKey.KEY_SHARE_TITLE	分享标题	否
ParamsKey.KEY_SHARE_DESCRIPTION	分享内容描述	否
ParamsKey.KEY_SHARE_TEXT	分享的文字	否
ParamsKey.KEY_SHARE_TARGETURL	分享点击跳转链接	是
ParamsKey.KEY_SHARE_IMAGEURL	分享图片链接	是

推送模块

一、开启推送

sdkObj.Push.startWork(Activity activity);

参数	参数说明
activity	游戏当前activity对象

二、设置推送tag

sdkObj.Push.setTags(String tag)

参数	参数说明
tag	需要设置的tag