

搜狐游戏中心 SDK For Android 参考手册

V1.0.0

2015/4/13

时间	版本	更新说明
2015/4/13	V1.0.0	1. 支持手机号码注册，登录； 2. 支持支付宝支付，微信支付，财付通支付。

目录

1. 前置条件	4
1.1 渠道号	4
1.2 环境	4
1.3 参数	5
2. SDK 配置	5
2.1 Application 配置	5
2.2 AndroidManifest.xml 配置	6
2.3 CP 参数设置	6
2.4 横竖屏设置	7
3. API 接入说明	8
3.1 接入次序	8
3.2 回调说明	8
4. API 功能介绍	9
4.1 初始化	9
4.2 登陆	10
4.3 是否登陆	11
4.4 销毁	12
4.5 注销(切换账户)	12
4.6 支付	13
4.7 获取登陆用户 ID	15
4.8 账号切换	16

1. 前置条件

1.1 渠道号

渠道号由搜狐游戏中心统一分配，最终渠道包由搜狐游戏中心产出。

1.2 环境

- 开发包支持在 Android SDK2.2 及以上固件版本运行,编译时请用 Android SDK4.0 以上版本编译。
- SDK中LibProject 工程里面的资源(即 res 文件夹下所有文件包括 drawable,id,string,color,layout 等)均以sohu_开头命名。CP 接入时请不要以这些为开头命名资源, 以避免冲突。

1.3 参数

接入 SDK 必须确保获得由搜狐游戏中心分配好的以下参数：

- AppId：游戏 APPID，初始化 SDK 使用；
- AppSecret：初始化 SDK 使用；
- productName：游戏名称，初始化 SDK 使用。

2. SDK 配置

2.1 Application 配置

- 如果您的游戏没有需要继承 Application 的类,那么您需将
“com.sohu.game.pay.SohuGameSDKApplication ” 作为
您 游 戏 的 Application 配置到 AndroidManifest.xml 里
面,如下:

```
<application
    android:name="com.sohu.game.pay.SohuGameSDKApplication"
```

- 如果您的游戏有需要继承Application的类,可让该类直接继承“SohuGameSDKApplication”,然后再配置到AndroidManifest.xml 里。实 现举例如下:

```
public class MyApplication extends
SohuGameSDKApplication {

    @Override
    public void onCreate() {
        super.onCreate();
        // 这里实现你的逻辑
    }
}
```

2.2 AndroidManifest.xml 配置

请将 Demo工程中的AndroidManifest.xml文件中的相关 permission, activity, service, receiver 等拷贝到您游戏 AndroidManifest.xml。

2.3 CP 参数设置

请将搜狐游戏中心分配给您的参数分别配置到 Manifest 文件中，如下：

```
<meta-data
    android:name="SOHU_GAME_SDK_APP_ID"
    android:value="@string/sohu_game_sdk_app_id" />
<meta-data
    android:name="SOHU_GAME_SDK_SECRET_KEY"
    android:value="@string/sohu_game_sdk_secret_key" />
<meta-data
    android:name="SOHU_GAME_SDK_PRODUCT_NAME"
    android:value="@string/app_name" />
<meta-data
    android:name="SOHU_GAME_SDK_LANDSCAPE"
    android:value="0" />
```

注意：以上配置需要介于 XML 文件的 `<Application>` `</Application>` 之间。

2.4 横竖屏设置

```
<meta-data
    android:name="SOHU_GAME_SDK_LANDSCAPE"
    android:value="0" />
```

要使用横屏请将 value 设置为 1，否则就为竖屏模式。

注意：以上配置需要介于 XML 文件的 `<Application>` `</Application>` 之间。

3. API 接入说明

3.1 接入次序

- 首先必须调用初始化接口；
- 对于有依赖于登录用户的接口如支付,获取用户 UID 等需要先调用登录 接口登录,您需要在登录接口的回调及设置切换账号结果通知回调里面 实现游戏的登录相关逻辑,具体请查看对应接口的功能介绍；
- 在程序结束的时候必须调用销毁接口，释放 SDK 相关资源。

3.2 回调说明

ISDKMethodResult<E>：回调接口

E: 接口状态码

在SDK提供的API中,有一部份API带有回调接口,如:初始化,登录,支付等。该回调接口为 ISDKMethodResult.Listener<T extends ISDKMethodResult>。CP需要实现其中的onResult (T result)方法。

T: 结果对象

CP可以通过result.getResultCode()来获取状态码，也可以快捷的使用result.assertResultCode(E)快速判断状态码。

根据该状态码实现自己业务逻辑接口状态码在下面API功能介绍时的举例内容里面都有定义,详细请查看各个 API 的功能介绍。

4. API 功能介绍

4.1 初始化

接口：

```
public void initialize(Activity activity,
    ISDKMethodResult.Listener<ISDKInitializeResult> listener);
```

状态码及描述：

状态码	描述
SUCCESS	初始化成功
CHECK_REQUIREMENTS_FAIL	环境检查失败
NETWORK_ERROR	网络错误
FAIL	其他错误

调用举例：

```
// 初始化
SohuGameSDK.initialize(this, new
ISDKMethodResult.Listener<ISDKInitializeResult>() {
    @Override
    public void onResult(ISDKInitializeResult result) {
        // 初始化成功
        if
(result.assertReturnCode(ISDKInitializeResult.RET_CODE.SUCCESS)
) {
            initDone();
        } else {
            Toast.makeText(MainActivity.this, "初始化失败，
"+result.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
});
```

4.2 登陆

接口：

```
public boolean login(Activity activity,
ISDKMethodResult.Listener<ISDKLoginResult> listener);
```

状态码及描述：

状态码	描述
SUCCESS	登陆成功
FAIL	登陆失败
CANCEL	用户取消

调用举例：

```
// 登陆
SohuGameSDK.login(MainActivity.this, new
ISDKMethodResult.Listener<ISDKLoginResult>() {
    @Override
    public void onResult(ISDKLoginResult result) {
        // 登陆成功
        if
(result.assertReturnCode(ISDKLoginResult.RET_CODE.SUCCESS)) {
            Intent i = new Intent(MainActivity.this,
GameActivity.class);
            startActivity(i);
            finish();
        } else if
(result.assertReturnCode(ISDKLoginResult.RET_CODE.FAIL)){
            Toast.makeText(MainActivity.this, "登陆失败.",
Toast.LENGTH_LONG).show();
        } else if
(result.assertReturnCode(ISDKLoginResult.RET_CODE.CANCEL)){
            Toast.makeText(MainActivity.this, "您取消了登陆请求.",
Toast.LENGTH_LONG).show();
        }
    }
});
```

4.3 是否登陆

接口：

```
public boolean isLoggedIn(Context context);
```

4.4 销毁

接口：

```
public void destory(Activity activity);
```

4.5 注销(切换账户)

接口：

```
public boolean exit(Activity activity,  
ISDKMethodResult.Listener<ISDKExitResult> listener);
```

状态码及描述：

状态码	描述
SUCCESS	成功
FAIL	失败

调用举例：

```
SohuGameSDK.exit(GameActivity.this, new
ISDKMethodResult.Listener<ISDKExitResult>() {
    @Override
    public void onResult(ISDKExitResult result) {
        if
(result.getReturnCode().equals(ISDKExitResult.RET_CODE.SUCCESS)
) {
            finish();
        } else {
            Toast.makeText(GameActivity.this, "注销失败",
Toast.LENGTH_SHORT).show();
        }
    }
});
```

4.6 支付

4.6.1 普通支付

接口：

```
public static boolean pay(Activity activity, IPaymentOrder
order, ISDKMethodResult.Listener<ISDKPayResult> listener)
```

状态码及描述：

状态码	描述
SUCCESS	支付成功
FAIL	支付失败

ORDER_FAIL	获取订单号失败
CANCEL	取消支付

调用举例：

```
String cpOrderId = UUID.randomUUID().toString();
SohuPaymentOrder order = new SohuPaymentOrder(
    PAY_AMOUNT,    // 支付金额（分）
    "新手礼包-998钻石", // 商品名称
    "1",           // 商品ID
    cpOrderId      // 穿透参数，CP自己的订单ID，用于后续通讯使用
);
SohuGameSDK.pay(GameActivity.this, order, new
ISDKMethodResult.Listener<ISDKPayResult>() {
    @Override
    public void onResult(ISDKPayResult result) {
        if
(result.assertReturnCode(ISDKPayResult.RET_CODE.SUCCESS)) {
            Toast.makeText(GameActivity.this, "支付成功",
Toast.LENGTH_SHORT).show();
        } else if
(result.assertReturnCode(ISDKPayResult.RET_CODE.ORDER_FAIL)) {
            Toast.makeText(GameActivity.this, "订单号获得失败",
Toast.LENGTH_SHORT).show();
        } else if
(result.assertReturnCode(ISDKPayResult.RET_CODE.FAIL)) {
            Toast.makeText(GameActivity.this, "支付失败",
Toast.LENGTH_SHORT).show();
        } else if
(result.assertReturnCode(ISDKPayResult.RET_CODE.CANCEL)) {
            Toast.makeText(GameActivity.this, "取消支付",
Toast.LENGTH_SHORT).show();
        }
    }
}):
```

4.6.2 微信支付

接口：

```
public boolean exit(Activity activity,
ISDKMethodResult.Listener<ISDKExitResult> listener);
```

第一步：获得 appId。信息提供，所提供信息包括：**应用中文名称**，**应用英文名称**（可选），**应用简介**，**应用英文简介**（可选），**应用图片**（两套，规格分别为 28*28，108*108），**应用官网地址**，**应用下载地址**（可选），**应用包名**，**应用签名**。具体可参考微信开放平台创建应用所需提供的应用信息。

第二步：接入开发。在工程配置文件里面配置搜狐游戏中心分配的 appId，并参考 Demo 工程接入微信支付。

说明：

1.接入文档补充：微信接入工作，请大家重点参考微信开放平台接口文档：
http://pay.weixin.qq.com/wiki/doc/api/app.php?chapter=8_5

2.回调说明：微信支付回调的接口不在 SohuGameSDK.pay 的回调方法里，而是在包名的 wxapi 包下的 WXPAYEntryActivity 类的 void onResp(BaseResp resp)里，请在此处添加对微信支付状态的处理。（具体状态请参考搜狐游戏中心提供的 Demo 工程）

4.7 获取登陆用户 ID

接口：

```
public String getLoginUid();
```

4.8 获取服务器 SessionKey

接口：

```
public String getSessionKey();
```

4.9 账号切换

接口：

```
public static void  
setAccountChangeListener(ISDKAccountChangeListener listener)
```

状态码及描述：

状态码	描述
SUCCESS	切换成功
FAIL	切换失败
CANCEL	取消切换

调用举例：

```
SohuGameSDK.setAccountChangeListener(new
ISDKAccountChangeListener() {
    @Override
    public void onAccountChangeListener(RET_CODE code) {
        if(RET_CODE.SUCCESS.equals(code)){
            //切换账号成功 此时为新账号登录状态
        }else if(RET_CODE.FAIL.equals(code)){
            // 切换账号失败 此时为注销状态
        }else if (RET_CODE.CANCEL.equals(code)){
            //取消切换 此时为前一个账号登录状态 或者 注销状态
        }
    }
});
```