统计 SDK 平台接入文档规范(网游）

V1.3.2

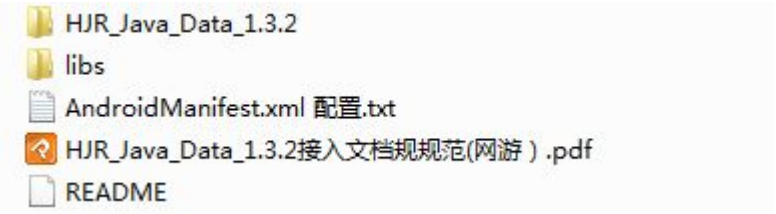
2015-04-23

更新记录			
版本号	**日期**	**更新内容**	**修订**
V1.2.0	2014-12-25	创建文档	张瑶
V1.2.0	2015-02-10	修改完善所有接口的接入点说明	张瑶
V1.3.0	2015-03-05	一、统计接口调整： 去掉（除登录接口，创建角色接口）其他接口重复的字段:用户唯一标识，服务器 ID,角色唯一标识 二、修改 AndroidManifest.xml： 1.将 meta 节点 name 为 statistic_serverurl 去掉，添加为 HJR_DATA_URL_DEBUG 模式 2.将 meta 节点 name 为 gamekey 改为 HJR_GAMEKEY 3.将 meta 节点 name 为 channel 改为：HJR_CHANNEL 三、所有统计数据接口添加重发机制	张瑶
V1.3.1	2015-04-10	1.更新优化，修复 bug	张瑶
V1.3.2	2015-04-23	1.更新优化，修复 bug 2.添加权限：<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS" />	张瑶

目录

一．SDK 构成

HJR_Java_Data_1.3.2 的资源如下图：



上图的资源说明：

1. HJR_Java_Data_1.3.2：统计sdk测试demo
2. libs：工程依赖的 libs 包,同HJR_Java_Data_1.3.2 里面的libs里面的文件，解压后将此文件夹里面的内容拷贝到游戏工程的相同libs目录下
3. AndroidManifest.xml 配置.txt ： AndroidManifest.xml的配置说明文件
4. HJR_Java_Data_1.3.2 接入文档规规范.pdf:统计sdk接入说明文档
5. README：更新sdk版本时需要修改的地方

二、SDK 植入步骤

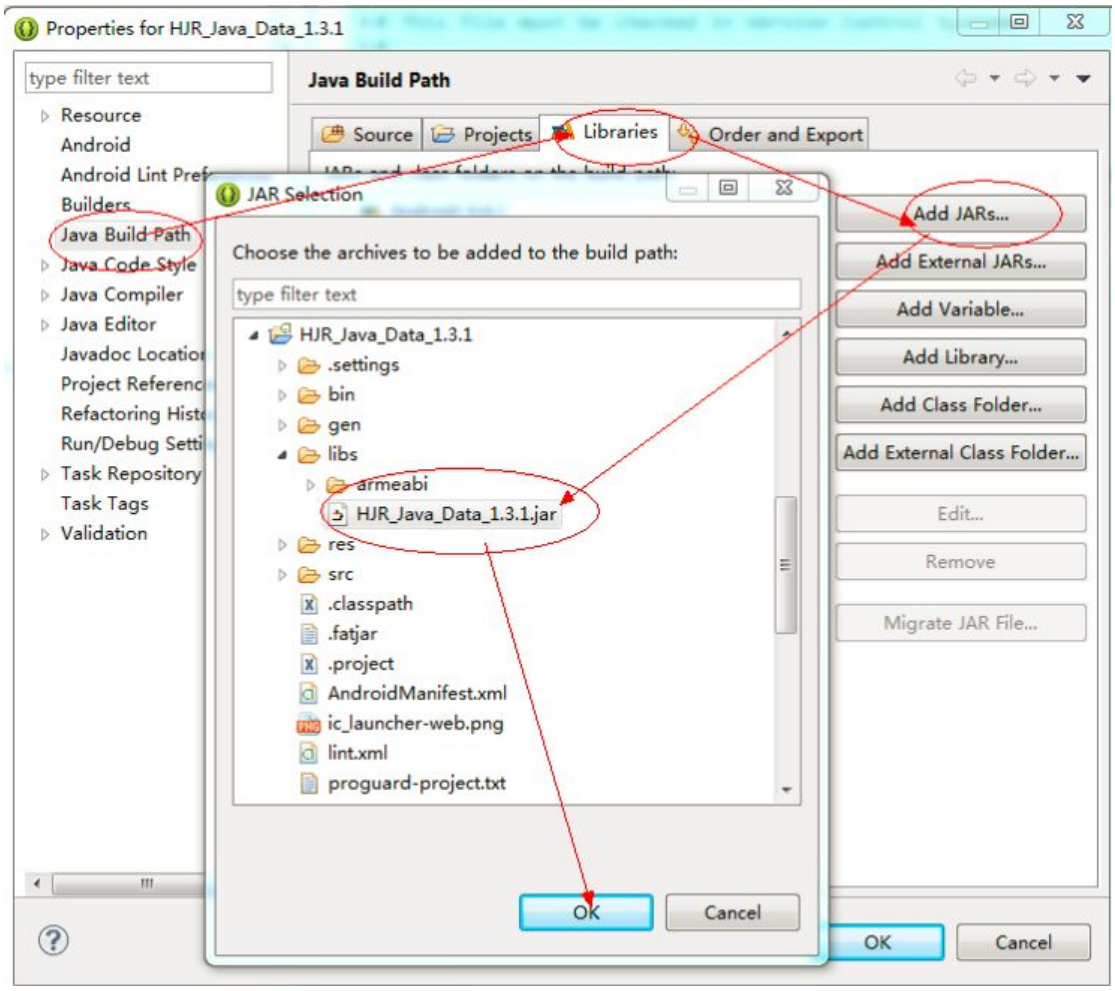
下面以 1.3.1 的版本举例，其他版本类似

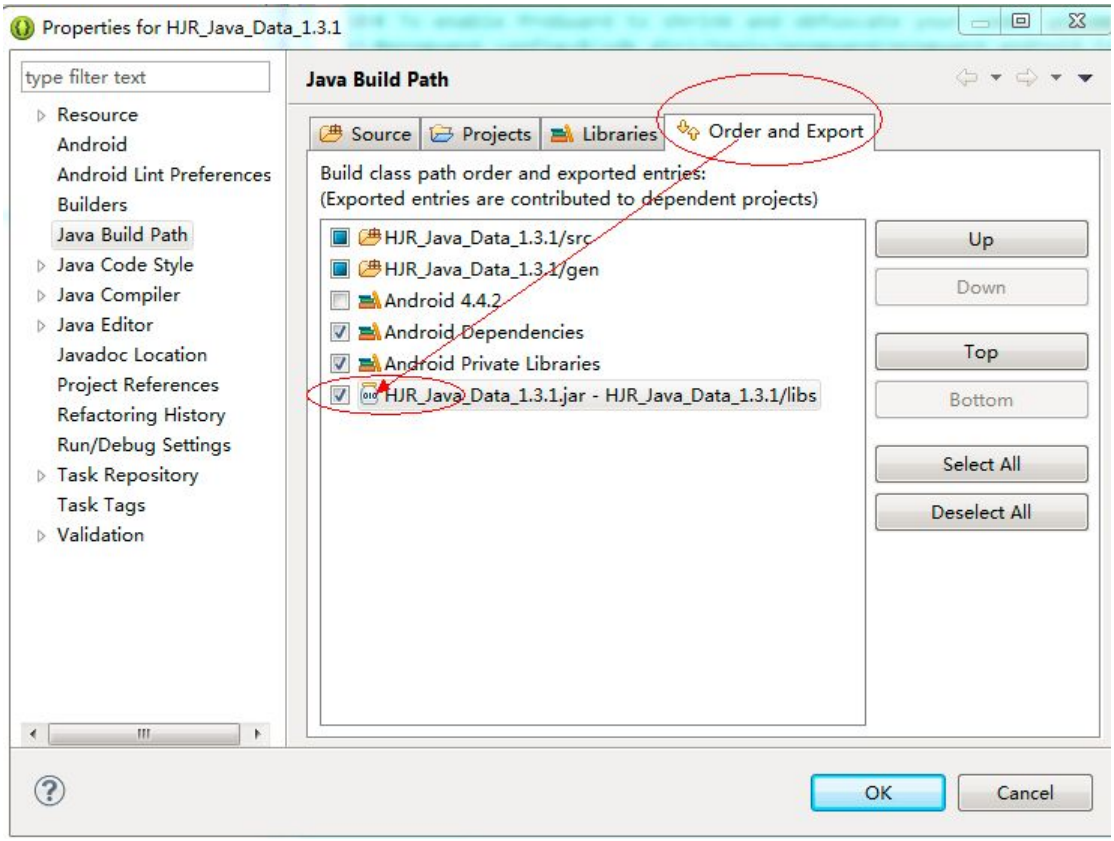
第一步：植入sdk前，请确保到平台下面已经申请好游戏密钥（gamekey）和下载最新的统计sdk版本了

第二步：解压sdk包 ，进入 libs ，将libs里面的所有资源(即下图所示的资源）拷贝到游戏工程的 libs 目录下

第三步：配置构建路径。Eclipse 用户右键工程根目录,选择 Properties -> Java Build Path -> Libraries,然后点击 Add External JARs... 选择指向 jar 的路径,点击 OK,在 Properties -> Java Build Path -> Order and Export 里面勾选中图中添加的 jar 包，点击 ok 即导入成功

=





注意： Eclipse ADT 17 以上版本用户,请在工程目录下建一个文件夹 libs,把 jar 包直接拷贝到这个文件夹下,刷新下 Eclipse 中工程就好了。 不要通过上述步骤手动配置构建路径的方式引入 Jar 包。 详情请参考 Dealing with dependencies in Android projects。

第四步：将“AndroidMainfest.xml 配置.txt”文件中的内容配置到游戏的AndroidMainfest.xml 中

配置说明：

1.在<application>节点外添加permission配置：

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS" />
```

2.在<application>节点下，添加 meta-data 配置：

```
<!-- 游戏密钥gamekey，需要游戏申请-->
<meta-data
    android:name="HJR_GAMEKEY"
    android:value="申请的gamekey" />


<!-- 渠道号    注意格式必须为\+空格+渠道号(flashAIR的游戏除外) -->
<meta-data
    android:name="HJR_CHANNEL"
    android:value="\ 自定义的渠道号" />

<!--测试环境请将DEBUG必须为true;正式上线请改为false，否则影响上线统计数据-->
<meta-data android:name="HJR_DATA_URL_DEBUG" android:value="true"/>
```

3.配置示例如下图：

注意：一定要在<application>节点下配置 meta-data 节点，否则会导致统计不上数据

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android_sdkkitsdk_statistic_test"
    android:versionCode="1"
    android:versionName="1.1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.PACKAGE_USAGE_STATS" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="HJR_Java_Data" >
        <meta-data
            android:name="HJR_GAMEKEY"
            android:value="ecb4d7dabc28315ce6d3d61e73499c66" />
        <meta-data
            android:name="HJR_CHANNEL"
            android:value="\ 0103" />
        <meta-data
            android:name="HJR_DATA_URL_DEBUG"
            android:value="true" />

        <activity
            android:name="com.example.sdkkit.statistic_test.MainActivity"
            android:configChanges="orientation|keyboardHidden|screenSize"
            android:screenOrientation="portrait" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

第五步：调用基本代码，在调用的Activity里面添加引用代码：

```
import com.sdkkit.gameplatform.statistic.SDKKitStatisticSDK;

import com.sdkkit.gameplatform.statistic.util.C;

import com.sdkkit.gameplatform.statistic.util.InitListener;

import com.sdkkit.gameplatform.statistic.util.ProtocolKeys;
```

（调用的接口具体详见下面：三、SDK 接口调用说明）

三、SDK 接口调用说明

1. SDK 初始化接口（必接）

接入点说明：在一个游戏启动的时候，即通常在启动的第一个 Activity 的 onCreate 方法里面调用

```
Map<String, Object> params = new HashMap<String, Object>() ;
//游戏类型 ： 1， 网游 ； 0 ， 单机
params.put(ProtocolKeys.KEY_GAMETYPE, "1");

SDKKitStatisticSDK.getInstance().init(this,params, new InitListener() {
                    @Override
                    public void onSuccess() {

                    System.out.println(" 初始化成功......");

                    }

                    @Override
                    public void onFailed() {

                    System.out.println(" 初始化失败......");

                    }
            });
```

2. 登录接口（必接）

接入点说明：在登录完成并且成功选择区服之后调用

```
Map<String, Object> params = new HashMap<String, Object>();

//必填，用户标识 ： 传入用户ID即可 ,*请注意：平台标识，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_USERMARK,"loginUserId");

//可选，整型(int(3))，用户类型：0为 临时用户，1为注册用户，2为第三方登录用户，游戏没有此字段的可不填
params.put(ProtocolKeys.KEY_USERTYPE,2);
```

```
//网游必填， 服务器编号,不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_SERVERNO, "10");

SDKKitStatisticSDK.getInstance().doUserLogin(params);
```

参数说明：

ProtocolKeys.*KEY_USERMARK*：用户标识(必填)，传入用户ID即可,*请注意：

ProtocolKeys.*KEY_USERTYPE*：用户类型（可选）,不能为中文或中文类型的特殊字符

ProtocoKeys.*KEY_SERVERNO* :服务器ID(网游必填) ,不能为中文或中文类型的特殊字符

备注：1. 用户标识实际的形式为：用户ID@平台 标识；其中平台标识，不能为中文或中文类型的特殊字符；@平台标识：可不填写，后台已自动拼接，以区分不同渠道平台的用户，对接方也可自定义填写，不影响统计

2. 所有的其他非本平台渠道的游戏,填写用户类型：

ProtocolKeys.*KEY_USERTYPE*，值为 2，均为第三方平台的用户

3. 创建角色接口（必接）

接入点说明：在游戏里用户登录选择区服进入游戏并创建角色成功之后调用

```
Map<String, Object> params = new HashMap<String, Object>();
//网游必填，角色唯一标识,不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ROLEMARK, "roleMark");

SDKKitStatisticSDK.getInstance().doCreateRole(params);
```

参数说明：

ProtocoKeys.*KEY_ ROLEMARK*： 角色标识(网游必填) ,不能为中文或中文类型的特殊字符

4．提交订单接口（必接）

接入点说明：1.在支付成功之后调用

（参数 ProtocolKeys.KEY_AMOUNT 的值以最终支付成功收到通知的实际金额为准，否则影响统计）

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，支付方式,可为中文
params.put(ProtocolKeys.KEY_PAYNAME, "支付宝");
//必填，充值金额（人民币，单位：元），不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_AMOUNT, "12");
//必填，订单号，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ORDERNUMBER, "11100");
//网游必填，整型(int),玩家等级
params.put(ProtocolKeys.KEY_UPGRADE,4);
//可选，商品描述,可为中文
params.put(ProtocolKeys.KEY_PRODUCT_DESC, "这里是我的商品描述");

SDKKitStatisticSDK.getInstance().doPostOrder(params) ;
```

参数说明：

ProtocolKeys.*KEY_PAYNAME*:支付方式(必填)，可为中文

ProtocolKeys.*KEY_AMOUNT*：充值金额(必填)，金额为人民币（单位：元），不能为中文或中文类型的特殊字符

ProtocolKeys.*KEY_ORDERNUMBER*：订单号(必填)，不能为中文或中文类型的特殊字符

ProtocolKeys.*KEY_UPGRADE*：玩家等级(网游必填），整型(int),

ProtocolKeys.*KEY_PRODUCT_DESC*：商品描述(可选)，可为中文

5．玩家升级接口（必接）

接入点说明：在登录游戏成功，并且成功选择区服后，当拿到用户的游戏等级信息时，请调用“玩家升级“接口

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，整型(int),玩家等级
params.put(ProtocolKeys.KEY_UPGRADE, 2);
SDKKitStatisticSDK.getInstance().doUserUpgrade(params);
```

参数说明：

PotocolKeys.*KEY_UPGRADE*：玩家等级(必填)，整型(int)

6．获得道具接口（必接）

接入点说明：在游戏里面获取道具之后调用

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，道具唯一标识，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMID, "3");
//必填，道具类型，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMTYPE, "2");
//必填，整型(int),道具数量
params.put(ProtocolKeys.KEY_ITEMCOUNT, 1);
//可选，原因,可为中文
params.put(ProtocolKeys.KEY_REASON, "原因");
SDKKitStatisticSDK.getInstance().doItemGet(params);
```

参数说明：

PotocolKeys.*KEY_ITEMID*：道具唯一标识(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMTYPE*：道具类型(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMCOUNT*：道具数量(必填), 整型(int)

PotocolKeys.*KEY_REASON*：原因(可选),可为中文

7．使用道具接口（必接）

接入点说明：在游戏里面使用道具之后调用

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，道具唯一标识，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMID, "3");
//必填，道具类型，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMTYPE,"2");
//必填，整型(int)道具数量
params.put(ProtocolKeys.KEY_ITEMCOUNT, 1);
//可选，原因，可为中文
params.put(ProtocolKeys.KEY_REASON, "原因");
SDKKitStatisticSDK.getInstance().doItemConsume(params);
```

参数说明：

PotocolKeys.*KEY_ITEMID*：道具唯一标识(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMTYPE*：道具类型(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMCOUNT*：道具数量(必填), 整型(int)

PotocolKeys.*KEY_REASON*：原因(可选),可为中文

8．购买道具接口（必接）

接入点说明：在游戏里面购买道具之后调用

```
Map<String, Object> params = new HashMap<String, Object>();

//必填，道具唯一标识，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMID, "3");
//必填，道具类型，不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_ITEMTYPE, "3");
//必填，整型(int),道具数量
params.put(ProtocolKeys.KEY_ITEMCOUNT, 1);
//必填，整型(int),虚拟币数量
params.put(ProtocolKeys.KEY_CURRENCYCOUNT,1);
//必填，虚拟币类型,可为中文
params.put(ProtocolKeys.KEY_CURRENCYTYPE, "1");
//可选，计费点，可为中文
params.put(ProtocolKeys.KEY_POINT, "");

SDKKitStatisticSDK.getInstance().doItemBuy(params);
```

参数说明：

PotocolKeys.*KEY_ITEMID*：道具唯一标识(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMTYPE*：道具类型(必填)，不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_ITEMCOUNT*：道具数量(必填), 整型(int)

ProtocolKeys.*KEY_CURRENCYCOUNT*：虚拟币数量(必填), 整型(int)，填写总数

PotocolKeys.*KEY_CURRENCYTYPE*：虚拟币类型(必填),可为中文

PotocolKeys.*KEY_POINT*：计费点(可选)，可为中文

9. 开始关卡接口（必接）

接入点说明：在游戏的每个关卡开始的时候调用

```
Map<String, Object> params = new HashMap<String, Object>();

//必填，玩家等级，整型(int)
params.put(ProtocolKeys.KEY_GRADE, 1);
//必填，关卡序号，整型(int)
params.put(ProtocolKeys.KEY_SEQNO, 1);
//必填，关卡唯一标识,不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_LEVELID,  "levelMark");
//必填，整型(int),关卡难度：0.简单，1.一般，2.困难，3.很困难，4.非常困难（以此类推，越难的排后，数字递增）
params.put(ProtocolKeys.KEY_DIFFICULTY, 1);


SDKKitStatisticSDK.getInstance().doStartLevel(params);
```

参数说明：

PotocolKeys.*KEY_GRADE*：玩家等级(必填)，整型(int)

PotocolKeys.*KEY_SEQNO*：关卡序号(必填)，整型(int)

PotocolKeys.*KEY_LEVELID*：关卡唯一标识(必填),不能为中文或中文类型的特殊字符

PotocolKeys.*KEY_DIFFICULTY*:关卡难度(必填), 整型(int)：0.简单，1.一般，2.困难，3.很困难，4.非常困难（以此类推，越难的排后，数字递增）

10．结束关卡接口（必接）

接入点说明：在游戏的每个关卡结束的时候调用

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，关卡序号，整型(int)
```

```
params.put(ProtocolKeys.KEY_SEQNO, 1);
//必填，关卡唯一标识,不能为中文或中文类型的特殊字符
params.put(ProtocolKeys.KEY_LEVELID,  "levelMark");
//必填，整型(int),关卡状态： 0.成功， 2失败，3退出 ；此处固定只能填0，或2，或3，填其他数字会导致没有统计数据或统计出错
params.put(ProtocolKeys.KEY_STATUS, 2);
//可选，原因（如失败点）,可为中文
params.put(ProtocolKeys.KEY_REASON, "失败点");
SDKKitStatisticSDK.getInstance().doEndLevel(params);
```

参数说明：

PotocolKeys.*KEY_GRADE*：玩家等级(必填)，整型(int)

PotocolKeys.*KEY_SEQNO*：关卡序号(必填)，整型(int)

PotocolKeys.*KEY_STATUS*：整型(int),关卡状态(必填)： 0.成功， 2 失败，3 退出 ；此处固定只能填 0，或 2，或 3，填其他数字会导致没有统计数据或统计出错

PotocolKeys.*KEY_REASON*：原因（如失败点）(可选),可为中文

11. 按钮点击接口（非必接）

接入点说明：在用户完成按钮点击之后，如：登录、进入游戏，充值等。

```
Map<String, Object> params = new HashMap<String, Object>();
//必填，按钮名称，可为中文
params.put(ProtocolKeys.KEY_NAME, "按钮名称");
SDKKitStatisticSDK.getInstance().doGameClick(params);
```

参数说明：

ProtocolKeys.*KEY_ NAME*：按钮名称(必填)， 可为中文

四、游戏在线时长统计及释放资源技术解决方案（必接）

如果你的工程中有Activity基类（例如BaseActivity），那么重写此基类的onStop()方法和onResume()方法，如下：

```
        /**游戏回到前台*/
    @Override
    protected void onResume() {
        super.onResume();
        if (!C.isActive) {
            SDKKitStatisticSDK.getInstance().saveFrontTime();
            C.isActive = true ;
        }
    }

    /**游戏进入后台*/
    @Override
    protected void onStop() {
            super.onStop();
      if (!SDKKitStatisticSDK.getInstance().isAppOnForeground()) {
        SDKKitStatisticSDK.getInstance().saveBackTime() ;
        C.isActive = false ;
          }
    }
```

如果你的工程中没有做 Activity 基类，那么你需要编写基类 BaseActivity，并且你的其他 Activity都要继承这个类，BaseActivity 如下：

```
public class BaseActivity extends Activity {
    @Override
    protected void onStop() {
```

```
        super.onStop();


        if (!SDKKitStatisticSDK.getInstance().isAppOnForeground()) {
              SDKKitStatisticSDK.getInstance().saveBackTime() ;
              C.isActive = false ;
              }
        }
        @Override
        protected void onResume() {
        super.onResume();
        if (!C.isActive) {
                    SDKKitStatisticSDK.getInstance().saveFrontTime();
                    C.isActive = true ;
              }
        }
}
```

（接入点说明：此功能是为了统计游戏在线时间，如果你的游戏中实现了统计在线时间的方法，那么不用做以上操作，只需要在你的游戏进入后台和回到前台的方法中调用统计接口即可：

游戏进入后台调用：

```
SDKKitStatisticSDK.getInstance().saveBackTime() ;
```

游戏回到前台调用：

```
SDKKitStatisticSDK.getInstance().saveFrontTime();
```

五、捕获并处理全局异常技术解决方案（非必接）

如果你的工程中有继承了 Application 的类，那么在此类的 onCreate()方法中添加如下内容：

```
// 异常处理，不需要处理时注释掉这两句即可！
CrashHandler crashHandler = CrashHandler.getInstance();
```

```
// 注册 crashHandler

crashHandler.init(getApplicationContext());
```

如果你的工程中没有继承 Application 的类，那么在你工程主包名目录下建立一个继承 Application的类,名称为 App，并在 AndroidManifest.xml 中注册此 Application 类，即在<application>中添加 name属性，示例代码如下：

```
<application

        android:name="App"

        android:icon="@drawable/ic_launcher"

        android:label="@string/app_name″

        android:theme="@android:style/Theme.Light.NoTitleBar.Fullsc
    reen">

        </application>
```

App 类的调用示例代码如下：

```
    public class App extends Application {

        @Override

        public void onCreate() {

            super.onCreate();

            // 异常处理，不需要处理时注释掉这两句即可！

            CrashHandler crashHandler =
        CrashHandler.getInstance();

            // 注册 crashHandler

            crashHandler.init(getApplicationContext());

            }

}
```

六、SDK 对接注意事项

1.*HJR_GAMEKEY*:需要向统计 sdk 平台申请正式的 gamekey;

HJR_CHANNEL：注意格式必须为\+空格+渠道号(flashAIR 的游戏除外),不同渠道平台填写的渠道号以区分平台，这个不需要向统计 sdk 平台申请，自定义填入

HJR_DATA_URL_DEBUG：是否为测试模式，游戏正式上线之前为 true，正式上线请改为 false

2.请所有的接口调用处严格按照“接入点说明”接入,只有确保准确的统计数据，才能正式上线

3.所有接口的用户唯一标识 (ProtocolKeys.*KEY_USERMARK*)，其后台格式为：

用户 ID@平台唯一标识；

用户 ID:为用户登录之后获取的 userId

平台标识：为接入的不同渠道平台的唯一标识，此标识统一由后台自动拼接，不用自定义传入

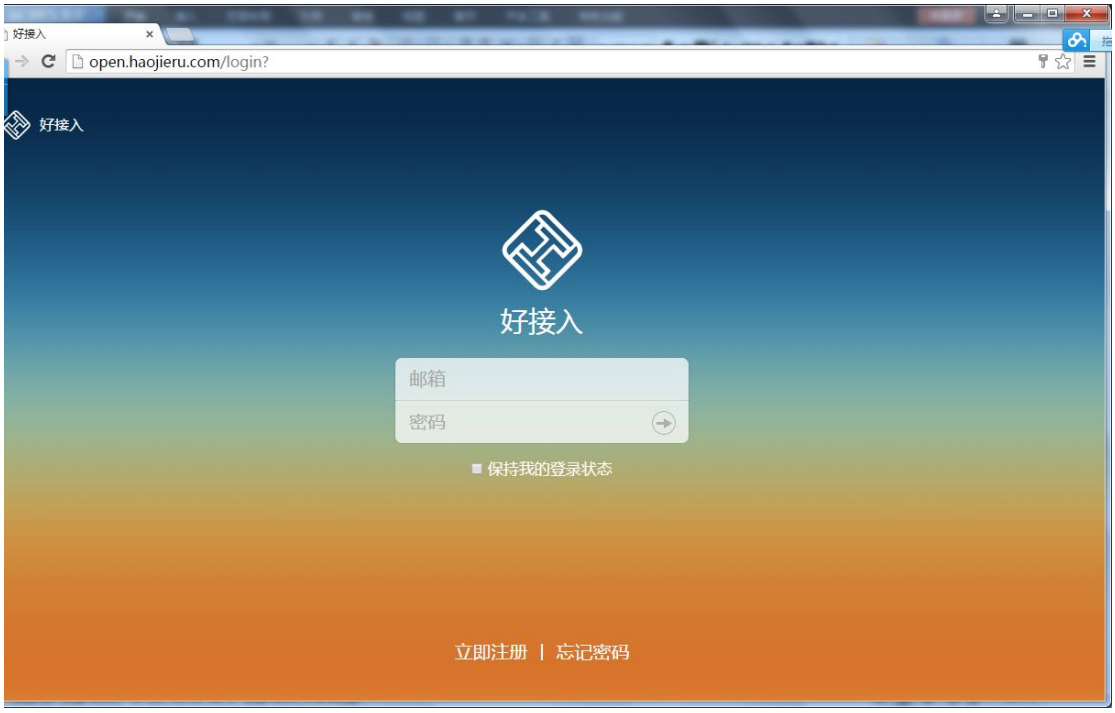
4.所有的必填项必须按照要求填写，尤其不能填写中文或中文字符的，填写了会导致统计出错

5.结束关卡接口中的状态必须按照固定整型 0,2,3 填写，否则统计出错

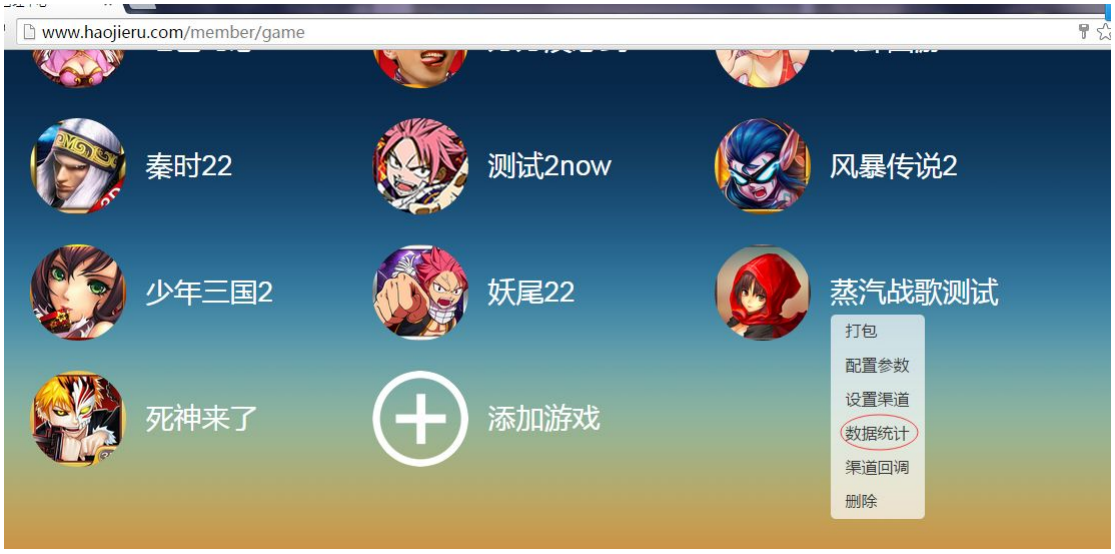
七、接入验证(两种方式，推荐方式一）

方式一：

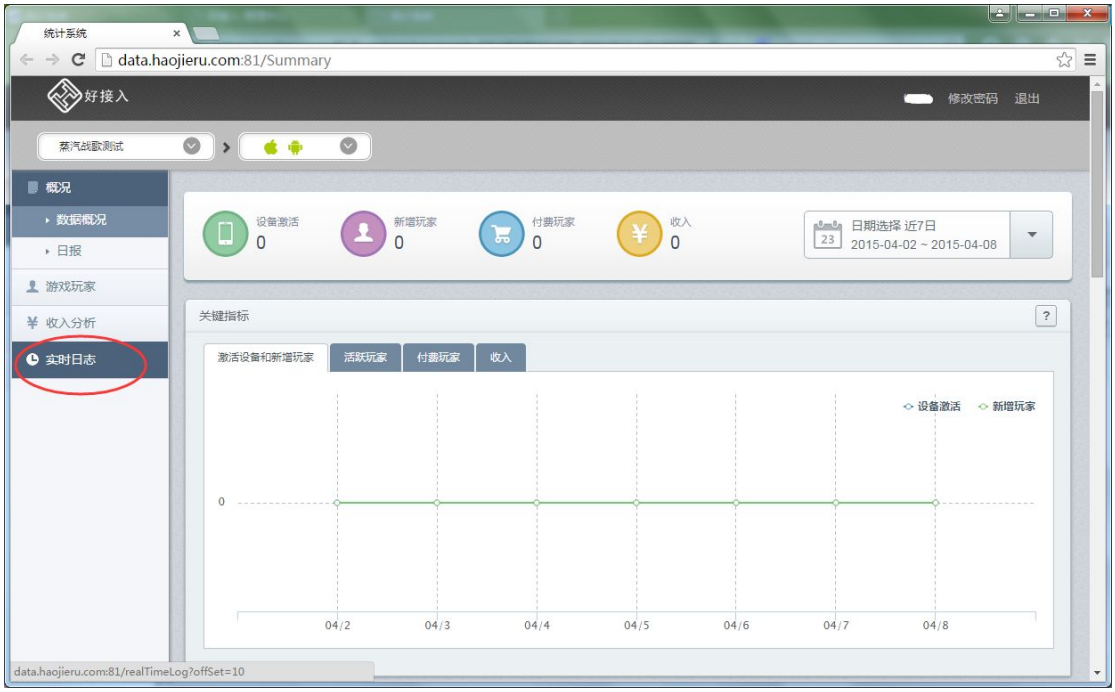
1.浏览器中输入地址：http://open.haojieru.com/，输入自己注册的账号密码登录，如下图所示：



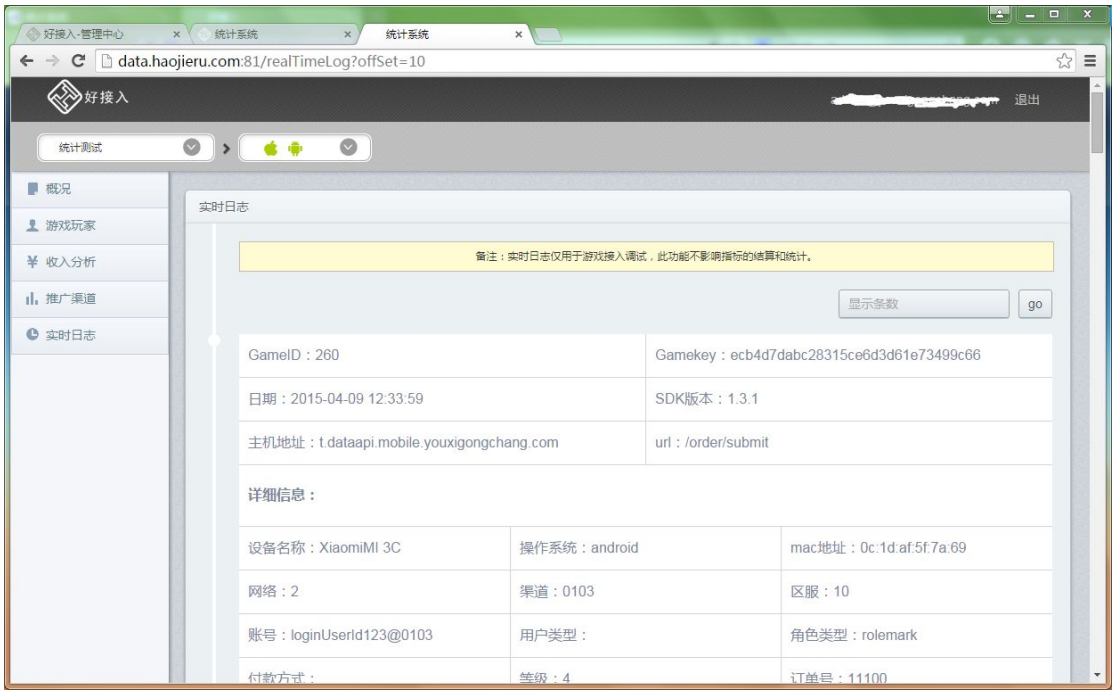
2.进入登录之后的页面，点击选中自己的游戏，出现下拉菜单，点击”数据统计 “，如下图所示：



3.进入游戏统计分析页面之后，点击”实时日志“选项，如下图所示：



4.进入自己的游戏统计数据测试页面，如下图所示：



方式二：

1.浏览器输入下面的地址：

测试分析页面（meta-data 里面的 HJR_DATA_URL_DEBUG 为 true 时)：
http://test.d.haojieru.com/test/analysis.php?gamekey=申请的 gamekey

正式分析页面（meta-data 里面的 HJR_DATA_URL_DEBUG 为 false 时)：

http://d.haojieru.com/test/analysis.php?gamekey=申请的 gamekey

例：分析页面之登录分析显示如下图：

```
[365] => Array
    (
        [date] => 2015-04-10 15:59:26
        [time] => 1428652766
        [session_id] => 6jjimnru0126pgg50go962s0c2
        [url] => /user/login/
        [host] => d.haojieru.com
        [ip] => 182.138.102.60
        [userid] => 0
        [gameid] => 260
        [info] => Array
            (
                [gamever] => 1.1.0
                [os] => android
                [single] => 1
                [wlan] => 0c:1d:af:c3:1a:ec
                [root] => 0
                [accept] => 0
                [osver] => 4.4.4
                [resolution] => 1920*1080
                [devicetoken] => 8bbd12d8007f99e
                [lang] => 2
                [network] => 2
                [version] => 1.3.1
                [cookie] => Y2hhbm5lbHwxfDE5MjAqMTA4MHxhbmRyb2lkNC4OLjR8WGlhb21pfDIuMHwwYzoxZDphZjpjMzoxlY3wyfENoaW5hIE1vYmlsZXxYaWFvbWlNSSAOV3xjbnx6aF9DTnx8MXx8MXwxLjEuMA==

                [gamekey] => ecb4d7dabc28315ce6d3d61e73499c66
                [source] => 2e82c4eba58760463338f2951f832265
                [device] => XiaomiMI 4W
                [channel] => 0103
                [cookie_decode] => channel|1|1920*1080|android4.4.4|Xiaomi|2.0|0c:1d:af:c3:1a:ec|2|China
            )

        [param] => Array
            (
                [serverno] => 10
                [filename_mark] => login_1428652765484
                [usertype] => 2
                [usermark] => loginUserId123@0103
                [success_flag] => 1
            )
```

分析页面需要关注的接口和对应的 url 参考如下：

1） 登录——/user/login/

2） 创建角色——/user/createrole/

3） 提交订单——/order/submit/

4） 游戏按钮点击——/sys/gamebtnclick/

5） 玩家升级——/user/upgrade/

6） 购买道具——/item/buy/

7） 获得道具——/item/get/

8） 使用道具——/item/consume/

9） 开始关卡——/level/pass/

10） 结束关卡——/level/pass/

11） 在线时长——/sys/gamedeactivate（acitivity 生命周期自动调用的）

前 11 个为接入验证的接口，下面的 12-14 号接口为后台自动调用的接口

12） 游戏激活——/sys/gamestart/（初始化后台自动调用）

13） 公网 ip——/sys/net/（初始化后台自动调用）

14） 异常记录——/sys/abnormal（程序发生异常时后台自动调用，可看到程序异常日志代码）

注意：所有的接口 14 除外：只有显示“操作成功”才算调用成功