

# 海游分析SDK

---

## 海游分析SDK

- 一、基本信息
- 二、快速对接
  - 1. 准备工作
  - 2. build.gradle 配置
  - 3. 配置 AndroidManifest.xml
- 三. 使用
  - 1、初始化
  - 2、注册 (LSDK)
  - 3、支付成功统计 (PSDK)
  - 4、appsfly支付验证 (PSDK)
  - 5、开始支付 (PSDK)
  - 6、支付失败 (PSDK)
  - 7、用户登录(LSDK)
  - 8、用户等级设置 (PSDK)
  - 9、用户特定等级统计
  - 10、游戏消费
  - 11、游戏进度上报
  - 12、完成新手引导上报
  - 13、创建角色
  - 13、解锁成就
  - 13、创建公会
  - 13、加入公会
  - 14、自定义事件

## 一、基本信息

---

- 版本 3.2.0
- 分析: (第三方分析 : 包含 appsflyer、talkingdata、facebookAnalysis)

## 二、快速对接

---

### 1. 准备工作

1. 复制 libs 中的文件到自己的项目 libs 中

### 2. build.gradle 配置

```
minSdkVersion 17 (大于等于17)

// aar 必须配置的
repositories {
    flatDir {
        dirs 'libs'
```

```

    }
}

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

    // 海游 SDK xx用版本号代替
    implementation(name: 'haiyou_common_xx', ext: 'aar')
    implementation(name: 'haiyou_analysis_xx', ext: 'aar')

    implementation(name: 'installreferrer-1.0', ext: 'aar')
    implementation(name: 'facebook-core', ext: 'aar')
    implementation(name: 'facebook-common', ext: 'aar')
    implementation(name: 'cardview-v7-28.0.0', ext: 'aar')
    implementation(name: 'haiyou_common_2.5', ext: 'aar')
    implementation(name: 'haiyou_analysis_2.5', ext: 'aar')
}

```

### 3. 配置 AndroidManifest.xml

- 权限声明

```

<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" />

```

- String文件声明FacebookID

```

<string name="facebook_app_id">[APP_ID]</string>
<string name="fb_login_protocol_scheme">fb[APP_ID]</string>

```

- Application声明

```

    <!-- 配置facebook-->
    <meta-data
        android:name="com.facebook.sdk.ApplicationId"
        android:value="@string/facebook_app_id" />
    <!--自动事件记录开关-->
    <meta-data
        android:name="com.facebook.sdk.CodelessDebugLogEnabled"
        android:value="true" />

    <provider
        android:name="com.facebook.FacebookContentProvider"
        android:authorities="com.facebook.app.FacebookContentProvider【FB-ID】

```

```

        android:exported="true" />

<activity
    android:name="com.facebook.FacebookActivity"
    android:configChanges="keyboard|keyboardHidden|screenLayout|screenSize|orientation"
    android:label="@string/app_name" />
    <!-- 配置facebook完成-->

    <!-- appsflyer 开始-->
    <receiver
        android:name="com.appsflyer.SingleInstallBroadcastReceiver"
        android:exported="true">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
    </receiver>
    <!-- appsflyer 结束-->
    <receiver
        android:name="com.walletfun.common.HaiYouReceiver"
        android:exported="true">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER" />
        </intent-filter>
    </receiver>

```

## 三. 使用

### 1、初始化

```

/*
 *启动 直接调用初始化 setAnalysisKey 需要在init前调用
 *@appsfly_DEVKEY:    appfly DEVKEY值
 *@talkingData_Key :  talkingdata 的 APPID
 *@FacebookId :      facebook 的 APPID
 */

HaiYouAnalysisSDK.instance().setAnalysisKey(this, appsfly_DEVKEY, talkingData_Key,
FacebookId);

walletHelp.addSDK(HaiYouAnalysisSDK.instance());

// 初始化 参数: context AppID key: 秘钥 sandbox:沙盒开关 ()
walletHelp.init(Context context, String APPID, String Key,
    int walletHelp.SANDBOX_CLOSE);

```

## 2、注册 (LSDK)

如果使用了HaiYou\_Login SDK 再内部已经调用 可以不做处理

统计的事件名为 completed\_registration

```
/**
 * 用户注册
 *
 * @param userID  用户ID
 * @param platform 用户登录方式
 */
HaiYouAnalysisSDK.instance().user_registration(Context context,
                                                String userID,
                                                String platform)
```

## 3、支付成功统计 (PSDK)

如果使用了HaiYou\_Pay SDK 再内部已经调用 可以不做处理

```
/*
 支付成功后调用
 * @orderId :  订单ID 确保每次订单的orderId唯一
 * @iapId :  商品名称
 * @currencyAmount :  现金金额或现金等价物的额度
 * @currencyType :  货币类型。例：人民币 CNY；美元 USD；欧元 EUR
 * @paymentType :  支付的途径，最多16个字符。例如：“支付宝”、“苹果官方”、“XX 支付SDK*/

HaiYouAnalysisSDK.instance().Pay_Suc(Context context
                                     ,String orderId,
                                     String goodsName,
                                     double currencyAmount,
                                     String currencyType,
                                     String paymentType);
```

## 4、appsfly支付验证 (PSDK)

APPSFLY 支付验证 在支付页面的OnActivityResult中调用

```
/**
 * Appsflyer google支付验证并上报
 *
 * @param activity
 * @param requestCode  onActivityResult 返回参数
 * @param resultCode  onActivityResult 返回参数
 * @param data        onActivityResult 返回参数
 * @param publicKey   google支付验证KEY值
 * @param currencyAmount 支付金额数
```

```

        * @param currencyType    货币单位
    */
    HaiYouAnalysisSDK.instance().Pay_AppsFlyPayVerify(Context context,
                                                    int requestCode,
                                                    int resultCode,
                                                    Intent data,
                                                    String publicKey,
                                                    String currencyAmount,
                                                    String currencyType) ;
}

```

## 5、开始支付 (PSDK)

用户发起支付订单时调用;

事件名: initiated\_checkout

```

/**
 * 开始支付 (SDK)
 *
 * @param orderID        订单ID 确保每次订单的orderID 唯一
 * @param goodsName      商品名称
 * @param currencyAmount 现金金额或现金等价物的额度
 * @param currencyType   货币类型。例: 人民币 CNY; 美元 USD; 欧元 EUR
 * @param paymentType    支付平台
 */
HaiYouAnalysisSDK.instance().Pay_checkout(Context context,
                                           String orderID,
                                           String goodsName,
                                           double currencyAmount,
                                           String currencyType,
                                           String paymentType)

```

## 6、支付失败 (PSDK)

用户支付失败或支付取消时调用

事件名: purchase\_cancelled

```

/**
 * 支付取消 (SDK)
 *
 * @param orderID        支付订单号
 * @param paymentType    支付平台
 */
HaiYouAnalysisSDK.instance().Pay_cancelled(Context context,
                                           String orderID,
                                           String paymentType)

```

## 7、用户登录(LSDK)

用户登录成功时调用

事件名: login\_x (x代表platform)

```
/**
 * 用户登录(SDK)
 *
 * @param userID
 * @param platform 登录方式
 */
HaiYouAnalysisSDK.instance().user_loginIn(Context context,
                                           String userID,
                                           String platform)
```

## 8、用户等级设置 (PSDK)

```
/**
 * 设置用户等级 (SDK)
 *
 * @param level 等级
 */
HaiYouAnalysisSDK.instance().user_setLevel(Context context, int level)
```

## 9、用户特定等级统计

用户达到要求等级时进行统计

事件名: achieved\_level\_x (x为等级)

```
/**
 * 用户特定等级统计
 *
 * @param level
 */
HaiYouAnalysisSDK.instance().user_setLevel_x(Context context, String level)
```

## 10、游戏消费

当用户进行虚拟物品或虚拟货币消费时 进行上报 (如 元宝、体力、钻石等)

事件名: spent\_(spentName)

```
/**
 * 游戏消费
 *
 * @param spentName 消费内容(如 ingot)
 * @param spentCount 消费数量
 */
HaiYouAnalysisSDK.instance().user_spent_x(Context context,
                                           String spentName,
                                           int spentCount)
```

## 11、游戏进度上报

用户达到要求游戏进度时进行统计

事件名: `achieved_process_x` (x为进度名)

[illegible]

## 12、完成新手引导上报

事件名: completed\_tutorial

[illegible]

## 13、 创建角色

事件名: create\_character

[illegible]

## 13、 解锁成就

事件名: create\_character

[illegible]

## 13、创建公会

事件名: create\_group

```
/**
 * 创建公会
 *
 * @param groupName 公会名
 */
HaiYouAnalysisSDK.instance().user_createGroup(Context context, String groupName)
```

## 13、 加入公会

事件名: join\_group

[illegible]

## 14、自定义事件

[illegible]



