

统计 SDK 开发者接入指南 1.0.1

目录

统计 SDK 开发者接入指南 1.0.1	1
Step 1、向工程中导入 SDK	2
Step 2、配置 AndroidManifest.xml 文件	2
Step 3、添加调用方法	3
Step 4、使用更多功能	4
渠道统计	4
自定义事件	5
异常错误信息报告	6
用户反馈	6
Step 5、检查并完成数据测试	7

STEP 1、向工程中导入 SDK

下载统计 SDK 压缩包 ([点击下载 SDK 包](#)) 并解压至本地目录, 将其中的 `gfansdk_statistics.jar` 导入到您的应用工程中。

以 Eclipse 为例, 其它开发工具导入外部 JAR 包请参照开发工具自身的用户手册:
右键点击工程根目录→选择 **Properties** --> **Java Build Path** --> **Libraries** → 点击 **Add External JARs...** 找到本地目录下的 `gfansdk_statistics.jar` 文件, 点击 **打开按钮** 即导入成功。



STEP 2、配置 ANDROIDMANIFEST.XML 文件

应用需要的权限	用途
INTERNET	允许应用程序联网, 发送统计数据的权限。
ACCESS_NETWORK_STATE	允许应用检测网络连接状态, 在网络异常状态下避免数据发送, 节省流量和电量。
READ_PHONE_STATE	允许应用以只读的方式访问手机设备的信息, 通过获取的信息来定位唯一的用户。
ACCESS_WIFI_STATE (可选)	用来获取设备的 mac 地址, 定位唯一的 android pad 用户。

meta-data	用途
gfan_statistics_appkey	用来唯一标识您的应用，将您获得的 gfan_statistics_appkey 写入 meta-data 中，用获得的 gfan_statistics_appkey 替换 “Your_gfan_statistics_appkey” 即可。
gfan_cpid (建议添加)	用来标注应用推广渠道，区分新用户的来源来查看统计，您可以使用 20 位以内的英文和数字为渠道定名，替换 value 中的 “Your_gfan_cpid” 。详见 渠道统计 。

示例代码：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest .....>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />

    <application .....>
        <activity .....>
            .....
        </activity>
        .....
        <meta-data android:name="gfan_statistics_appkey" android:value="Your_gfan_statistics_appkey" />
        <meta-data android:name="gfan_cpid" android:value="Your_gfan_cpid"/>
    </application>
</manifest>
```

STEP 3、添加调用方法

- ✧ 在需要调用 SDK 方法的类中导入 com.gfan.sdk.statistics.GFAgent 包。
- ✧ 对应用中每个 Activity 的 onResume 方法里调用 GFAgent.onResume(Activity context)，传入的参数为 Activity 上下文。

```
@Override
protected void onResume (){
    super.onResume();
    GFAgent.onResume (this);
}
```

- ✧ 在相应的 Activity 中的 onPause 方法里调用 GFAgent.onPause(Activity context)，传入的参数为当前的 Activity 的上下文。

```
@Override
protected void onPause(){
    super.onPause();
    GFAgent.onPause(this);
}
```

注 1：请确保您在每个的 activity 中都调用 GFAgent.onResume(Activity context)和 GFAgent.onPause(Activity context)方法，这将有效帮助您获取应用每个页面的统计信息，帮助分析应用中可能存在的体验问题。

注 2：对有继承或控制的 Activity 中，只需要对上层 Activity 添加调用即可，否则会重复计量。

STEP 4、使用更多功能

渠道统计

通过给不同的应用分发渠道（应用商店、下载站、论坛）提供不同的渠道标记包，可以在渠道统计中单独查询到他们的数据。同样可以对自己的官方网站、mobile 网站、论坛、某次活动打上渠道号，单独进行统计。

- ✧ 请在 AndroidManifest.xml 里添加 meta-data 段，用您定义的渠道名替换 value 中的 “Your_gfan_cpid”。
- ✧ 渠道名可自行定义，请使用 20 位以内的英文和数字。对以下流行渠道建议您使用下表中的建议渠道名。

```
<meta-data android:name="gfan_cpid" android:value="gfan.com"/>
```

- ✧ 不同的渠道 ID 包请重新编译打包。
- ✧ 每台设备仅记录首次安装激活的渠道，更替渠道包安装不会重复计量

建议的渠道名称：

渠道名称	meta-data 的 value
------	-------------------

默认渠道	default 或不填写
开发者官网	official website
Google Market	market.android.com
机锋网	gfan.com
安卓网	hiapk.com
安智网	goapk.com
N 多网	nduoa.com
优亿网	eoemarket.com

自定义事件

统计event_id对应事件的发生次数，变化趋势，例如广告点击，短信数量等等。参数context为当前Activity的引用。event_id为当前统计的事件ID，注意event_id中不要加空格或其他 的转义字符

- ✧ 在应用程序的 event 事件中加入下面格式的代码，也就成功的添加了一个简单的事件到您的应用程序中了：

`GFAgent.onEvent(Activity activity, String EVENT_ID);`

- ✧ 如果你想获取某些变量的值或者其它的一些数据，您可以按照下面的格式添加代码：

```
GFAgent.onEvent(Activity activity, String EVENT_ID , String  
EVENT_LABEL);
```

异常错误信息报告

通过收集应用程序的异常信息来帮助您完善自己的程序。您可以通过两种方式来向我们的服务器端发送异常错误信息。

- ✧ 一种是自动捕获异常信息。

只有给 `GFAgent.setReportUncaughtExceptions(Boolean enabled)` 方法传入 `true` 时,才会自动捕获异常信息 ;如果你想关闭自动捕获异常信息 ,您可以给该方法传入 `false`。

如下所示 :

```
@Override  
public void onCreate(Bundle savedInstanceState){  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    GFAgent.setReportUncaughtExceptions(true);  
}
```

- ✧ 另一种是开发者自己传递的错误信息。

您可以通过手动的调用 `GFAgent.onError(Activity activity, Throwable throwable)` ,然后把抛出的 `exception` 传入到该方法的第二个参数中。

如下所示 :

```
try {  
    File file = new File("filePath");  
    if(!file.exists()){  
        file.createNewFile();  
    }  
} catch (IOException e) {  
    GFAgent.onError(this, e);  
}
```

建议 : 如果您想更早的捕获用户触发的异常 , 请在程序最先启动的地方加入以下代码 :

```
TCAgent.init(this);
```

用户反馈

通过收集用户的反馈信息来完善您的应用程序。您可以通过以下步骤来集成用户反馈模块：

- ✧ 将 zip 包里的资源文件复制到您的应用程序 res 文件下，您可以更换里面的资源和样式，但不要更改资源文件的名称和控件的 ID。并在 AndroidManifest.xml 文件里注册 SendFeedback Activity。

```
<activity android:name="com.gfan.feedback.SendFeedback"
          android:windowSoftInputMode="adjustResize" />
```

- ✧ 在需要启动反馈界面的位置加入如下代码：

```
GFAgent.onStartFeedbackActivity(Activity activity);
```

STEP 5、检查并完成数据测试

检查以下环节：

- ✧ 确保 AndroidManifest.xml 里的参数和必要的权限已正确添加，再次检查 INTERNET, ACCESS_NETWORK_STATE, READ_PHONE_STATE 这些权限。
- ✧ 确保 App ID 添加正确。
- ✧ 确保每个 Activity 中的对 onResume 和 onPause 方法进行了正确调用。
- ✧ 如需要检测渠道包，确保 gfan_cpid 添加正确。
- ✧ 确保您的测试设备能够连接网络。
- ✧ 启动应用。
- ✓ 登录统计后台查看数据。