
科大讯飞股份有限公司

iFLYTEK CO.,LTD

科大讯飞统计分析 SDK 集成指南

1. 概述

通过集成讯飞开放统计 SDK 可以跟踪移动应用的用户属性及操作行为，您可以登入讯飞开放平台 www.xfyun.cn，进入 我的语音云-我的应用，开通应用的开放统计服务，即可查看、下载到数据处理的结果。阅读如下文档，您可以在几分钟之内轻松完成集成过程。

2. 集成说明

2.1 申请 appid

在 www.xfyun.cn 网站中注册成为开发者并创建一款应用，您将获得一款标识您的应用的 appid。

2.2 导入 SDK

1. 下载最新版 SDK 的 zip 包，将其中的 libs 文件夹下 **Sunflower.jar** 合并到本地工程 libs 子目录。
2. Eclipse 用户右键点击工程根目录，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，即导入成功。

2.3 配置 AndroidManifest 文件

打开 AndroidManifest.xml 文件，增加如下权限：

权限	用途
INTERNET	允许程序联网和发送统计数据的权限。
ACCESS_NETWORK_STATE	允许应用检测网络连接状态。
READ_PHONE_STATE	允许应用以只读的方式访问手机设备的信息，通过获取的信息来唯一标识用户。
ACCESS_WIFI_STATE	获取设备的 MAC 地址，同样用来标识唯一用户。
ACCESS_COARSE_LOCATION（可选）	获取设备的上传数据时的位置信息，提高统计精准度。
meta-data	用途
IFLYTEK_APPKEY	用来唯一标识您的应用，请将以下示例代码中 your_app_key 替换为您在讯飞后台申请的应用 appid。
IFLYTEK_CHANNEL	用来标注应用推广渠道，区分用户来源，请将以下示例代码中 your_channel 替换为您自定义的渠道名称。 格式：32 个字符以内，支持中文、英文、数字，请勿使用其他符号。

示例代码：

```
<manifest.....>
<application .....>
<activity ...../>
<!--请注意添加单引号-->
<meta-data android:name="IFLYTEK_APPKEY"android:value="'your_app_key'" />
<meta-data android:name="IFLYTEK_CHANNEL" android:value="your_channel"/>
</application>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
```

2.4 添加代码

在每个 Activity 的 onResume 方法中调用 FlowerCollector.onResume(Context)，在 onPause 方法中调用 FlowerCollector.onPause(Context) 如下：

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

说明：

1. 确保在所有的 activity 中都调用 FlowerCollector.onResume() 和 FlowerCollector.onPause() 方法。这两个调用将不会阻塞应用程序的主线程，也不会影响应用程序的性能。
2. 注意，如果您的 Activity 之间有继承或者控制关系请不要同时在父和子 Activity 中重复添加 onPause 和 onResume 方法，否则会造成重复统计(eg: 使用 TabHost、TabActivity、ActivityGroup 时)。
3. 一个应用程序在多个 activity 之间连续切换时，会被视为同一个 session(启动)。
4. 当用户两次使用之间间隔超过 30 秒时，将被认为是两个的独立的 session(启动)。例如：用户回到 home，或进入其他程序，经过一段时间后再返回之前的应用。
5. 所有日志收集工作均在 onResume 之后进行，在 onPause 之后结束。

2.5 数据收集接口（FlowerCollector）

1.功能设置接口

1、开启调试模式

FlowerCollector.setDebugMode(Boolean flag)

开启后可以在 logcat 中看到相应的日志，默认不开启。

2、设置会话时间，默认 30s（单位为毫秒）。

FlowerCollector.setSessionContinueMillis(long time)

3、开启页面统计模式

系统默认通过 OnResume 和 OnPause 统计页面，可以调用 FlowerCollector.openPageMode(true)开启页面统计模式。开启后，可以通过 onPageStart(String pageName) ,onPageEnd(String pageName) 自定义页面的名称。

4、开启自动获取位置信息

FlowerCollector.setAutoLocation（Boolean flag）

开启后会在每次发送日志时获取设备位置信息，默认开启。

5、开启自动捕获异常信息

FlowerCollector.setCaptureUncaughtException(Boolean flag)

开启后会在自动捕获程序的异常信息，默认关闭。

---个性化接口---

6、设置用户性别

FlowerCollector.setGender(Context context,Gender gender);

7、设置用户年龄

FlowerCollector.setAge(Context context, int age);

8、设置用户 id

FlowerCollector.setUserID(Context context, String userId);

注意：

- 1.以上设置接口需要在程序的入口处设置，例如：在 onCreate 方法中调用。
- 2.确保设置正确的 appid，否则会导致无法统计日志。
- 3.开启页面统计模式时请确保 onPageEnd(String pageName) 中 pageName 与 onPageStart 中一致。

```
@Override
protected void onResume() {
    super.onResume();
    FlowerCollector.onResume(this);
    FlowerCollector.onPageStart("NewsPage");
}

@Override
protected void onPause() {
    super.onPause();
    FlowerCollector.onPageEnd("NewsPage "); //请确保 onPageEnd 在 onPause 之前调用。
    FlowerCollector.onPause(this);
}
```

2.事件统计接口

2.1 在希望记录事件发生的地方调用如下方法:

`FlowerCollector.onEvent(Context context, String eventId)`

例如: 用户点击了一次播放音乐:

`FlowerCollector.onEvent(Context mContext, "music")`

如果您需要记录音乐的名称:

`FlowerCollector.onEvent(Context mContext,"music","爸爸去哪里哇")`

如果您还需要记录更详细的信息:

`HashMap<String, String> map = new HashMap<String, String>();`

`map.put("song", "爸爸去哪里哇");`

`map.put("type", "pop");`

`map.put("format", "mp3");`

`FlowerCollector.onEvent(this, "music",map);`

2.2 记录事件的持续时间 (单位毫秒)

1.自己计算,传入时间

`FlowerCollector.onEventDuration(Context context, String eventId, long duration)`

例如: 音乐播放了 5 分钟

`FlowerCollector.onEventDuration(mContext,"music", 5 * 60 * 1000);`

2.通过 SDK 计算

在事件开始时调用 `FlowerCollector.onEventBegin (Context context, String eventId)`

在事件结束时调用 `FlowerCollector.onEventEnd (Context context, String eventId)`

注意:

1.请确保事件开始和结束的 `eventId` 一致,在 Debug 模式下可以看到事件是否添加成功的 log。

2. `event_id` 和 `label` 不能使用特殊字符,且长度不能超过 128 个字节; `map` 中的 `key` 和 `value` 都不能使用特殊字符, `key` 和 `value` 均不能超过 128 个字节。

3.错误信息记录

1.用户可以通过调用 `OnError(Context context, String msg)` 传入程序的错误信息。

2.如果您的应用在使用过程中不幸发生崩溃,我们也会将错误信息自动记录下来以供您进行分析。实现此功能需要您在程序入口处调用 `setCaptureUncaughtException(true)`, 开启异常信息自动捕获功能。

4.立即上传日志

调用 `flush()`接口会立即发送一次日志,此功能可用作调试,不建议作为发送策略使用。

5.数据上报策略

您可以通过讯飞网站上配置日志上报的频率。目前支持: 启动时发送、按时间间隔发送。

需要在程序的入口中添加 `FlowerCollector.updateOnlineConfig(Context context,null)`, 用来获取后台配置的策略。默认启动时发送。

6.在线参数

您在讯飞网站上设置 Key-Value 值之后，可以调用如下接口动态获取线上最新的参数值：

1.从服务端获取参数、缓存本地：

```
FlowerCollector.updateOnlineConfig(mContext,new OnlineConfigListener() {  
    @Override  
    public void onDataReceived(JSONObject jsonobject) {  
        //回调仅在参数有变化时发生  
    }  
});
```

2.获取具体参数：

```
String value = FlowerCollector.getOnlineParams(mContext, "xxx");  
xxx 为您在网站上设置的 key。
```

2.6 混淆代码

工程混淆设置：如需打包或者生成 APK 的时候进行混淆，在 proguard.cfg 中添加如下代码：
*-keep class com.iflytek.sunflower.**{*};*

3. 查看数据

集成完成后，您可以登入讯飞开放平台 www.xfyun.cn，进入[我的语音云-我的应用](#)，开通应用的开放统计服务（如下图），即可查看、下载到数据处理的结果。

