

## Azard的博客

月球背面

# Android用户行为收集工具调查

📅 2016-01-12 | 📁 [Android](#) | 💬 | 📖 本文总阅读量 475 次

这篇博客介绍总结了国内外，面向Android开发者的用户行为收集、分析、错误崩溃信息收集反馈的开源、商业工具，以及他们的一些特点。

**Android用户行为收集工具** 是面向开发者，通常作为library嵌入在程序中的，将用户行为和程序错误信息反馈给开发者的工具。

从[友盟Demo](#)可以看到此类工具的作用，就是一部分集成在apk安装在用户手机里，当使用该应用的时候，会自动将用户的点击信息等各种信息发送到服务端，让开发人员根据数据调整应用，让测试人员获知在特定设备特定版本的程序错误，让运营人员调整运营模式。

国内做这方面工具的公司，主要有 [友盟](#)、[TalkingData](#)，还有专门做移动游戏分析的[DataEye](#)。国外主要有做用户行为分析的[Google Analytics for Android](#)，Yahoo收购的[Flurry](#)，做错误信息收集反馈的Twitter的[Fabric](#)，以及开源项目[ACRA](#)。

## 总体功能列表

作为用户行为分析收集工具，所谓 **session** 就是指记录用户启动应用，开启 Activity 页面的次数和起止时间，结合单一用户识别，能组合统计出许多功能，见[友盟Demo](#)，包括用户趋势，活跃度，用户回流率等等内容。

	友盟 Umeng	TalkingData	Flurry Analytics
客户端功能			
Activity session	√	√	√
Service session	×	×	√
自定义事件	√	√	√
自定义事件时长	×	×	√
热添加自定义事件	×	√	×
读Log Report	√	√	√
自定义Error Report	√	√	√
服务端功能			
实时变化	1小时	1小时	15秒
渠道统计	√	√	Google Play单一不需要
session相关统计	√ 丰富	√ 丰富	√ 丰富，可定制显示
用户趋势相关统计	√ 丰富	√ 弱	√ 丰富，可定制显示
地理位置分布	√	√	√

	友盟 Umeng	TalkingData	Flurry Analytics
版本分布	√	√	√
设备分布	√	√	√
页面访问路径	√	×	√ 可定制显示
用户画像	√ 基于分享功能	×	√ 基于各种技术
开放API	×	√	×
前端可定制	×	×	√

国内的两款商业产品从功能上比 Flurry 都要弱一些，其中 TalkingData 开放了API用户可以自己定制想要获取的数据，而Flurry将这种功能做成了前端可视化编程，对报表有很强的自定义性，使用起来比较方便。

## 友盟 Umeng

友盟Analytics工具的使用步骤可以见官网，主要是添加jar包依赖，manifest的元数据中写入

```
<meta-data android:value="YOUR_APP_KEY" android:name="UMENG_APPKEY"></meta-data>
<meta-data android:value="Channel ID" android:name="UMENG_CHANNEL"/>
```

分别用于识别唯一App和渠道分发途径。

然后在 Activity 的 onResume() 中添加 MobclickAgent.onResume(this)，在 onPause() 中添加 MobclickAgent.onPause(this) 用于侦测 Activity 的开启关闭。

这里稍微对友盟需要收集的数据做个猜想，移动端的session统计基本也就涵盖这些数据

- metadata-appkey：标记该应用。
- IMEI，mac，运营商：标记唯一用户，也许第一次用过后会生成一个hash值用来代替这两项。
- metadata-channel：标记渠道，第一次发送过后不再发送。
- 设备信息：只发送一次，设备信息在服务端和用户绑定。
- versionName：区分一个app不同的版本号。
- 联网方式：wifi或者3G 4G。
- IP：获取地域信息，也许是间隔天数发送该信息，一段时间可以在服务端绑定地域和用户。
- Activity信息：用于统计访问路径
- time：因为在离线的时候这些信息会保留在本地，所以应该是获取的本机时间。也许在线的时候时刻信息可以以服务端收到的时刻为准。根据onResume和onPause的时刻间隔可以当做使用时长。

猜测后端实现保留2天的小时为单位的数据统计结果，2天以上的数据转成以天为单位存储。

## TalkingData

### TalkingData Demo

TalkingData的使用方法和Umeng完全一样。但是自带报表的功能没有Umeng丰富，前端没Umeng做的漂亮。但是提供开放API通过JavaScript可以查询。

```
{
  "accesskey": "fb46c6980e294be483fa*****be35"// *填写应用的接入码
  "metrics":["newuser"],          // *填写要查询的指标项
  "groupby":"daily",              // *填写数据维度，即数据分组方式
  "filter":{                       // *数据筛选条件
    "start":"2015-04-14",          // *查询时期的起始日
    "end":"2015-04-15",           // *查询时期的截止日
  }
```

```
"platformids":[1,2],          //限定要查询的系统平台
"versions":["1.0","2.0"],      //限定应用版本号
"channelids":[1605,1607],      //限定查询渠道
"eventids":["clear","addCar"], //限定查询的事件id
"pagenames":["index","pay"]   //限定查询的页面名
},
"order":"desc",               // 数据结果排序方式，除按时间分组外默认倒序
"limit":10,                   // 限定返回数据的条数
"sum":true,                   // 返回结果中给出数据总和
"avg":false                   // 返回结果中给出数据的平均值
}
```

在[API调用的参数说明](#)中可以初步看出TalkingData的后端是如何设计的。

TalkingData有一个有趣的新功能，热更新自定义事件。他们官方叫做 **灵动分析**，他们写了相关的3篇文章放在[知乎专栏](#)上。

## Flurry

Flurry的session域测定的广一点，是在 `onStart()` 和 `onStop()` 之间，并且也能测定 `Service` 的session。

和Umeng、TalkingData最大的区别是报表可以定制化，而且定制化程度很高，比TalkingData的开放API好用，Umeng基本是死报表。见[YouTube视频](#)。

## ACRA

[ACRA](#) 是一个开源的专门针对Android Crash Report的工具，它的错误报告的功能比上面Analytics工具的bug report模块要强的多。

用法主要是把应用的 `Application` 传给ACRA进行各种黑科技操作，report可以自定义各种功能。

服务端有ACRA组织自己的实现，也有一些商业的实现。

## Fabric

Fabric是Twitter的移动开发工具套件，[全面介绍视频](#)，[文档](#)。

Fabric是一个比较成熟的工具库（比ACRA成熟很多），最主要的工具Crashlytics是个bug report工具，比较简单的用户行为分析的工具也有，还有很多团队工具，感觉上是面向一个团队开发和测试使用的工具链。

Fabric的使用方式比较特别，需要把依赖的 gradle 替换成Fabric专用的 gradle 。

```
apply plugin: 'io.fabric'
```

Fabric的功能比较全面，是一个完善的商业产品

- 自动对Crash Report分级
- web的board操作端有一些例如issues的便于团队合作的功能
- 有叫“[Beta Process](#)”的团队开发测试工具，可以通知测试人员最新的build
- App对版本分类
- 自带ProGuard，DexGuard功能
- 自定义一些log，bug report
- 自定义bug report sender
- NDK crash report（ACRA没有这个功能，Twitter客户端本身有很多NDK写的），可以同样做stack trace，custom log。

坚持原创技术分享，您的支持将鼓励我继续创作！

赏

# Android

---

◀ 再见2015，你好2016

ACRA源码分析 ▶

1

© 2013 - 2017 ♥ Azard

由 [Hexo](#) 强力驱动 | 主题 - [NexT.Pisces](#)

👁本站总访问量 11723 次

