Analytics

开发者接入指南 1.2

最后修订：2014-08-13

目录

一、 综述

1. 适用范围

TalkingData Analytics 为移动应用提供数据统计分析服务，通过在应用中加入统计 SDK，来在 Analytics.TalkingData.net 网站中查阅应用的相关数据。

SDK 适用于 iOS5.1.1 及以上操作系统的设备。

2. 统计标准

数据系统中的基本数据单元依据以下标准定义：

- 新增用户

  TalkingData 数据系统中的“用户”指用户的一台唯一设备。

- 用户的一次使用

  指用户从打开应用的界面至离开界面的完整过程。如用户在离开界面后 30 秒内重新回到应用中 ,将被认为是上次使用被打扰后的延续 ,记为一次完整使用。

- 页面

  可以认为是应用在手机界面显示的 Page view ,由开发者调用 trackPageBegin 和 trackPageEnd 来定义一个页面的开始和结束，页面不能嵌套。

- 自定义事件

  指用户在应用中进行了特定的操作或达成了特定的条件。例如 : 用户点击了广告栏、用户进行付费等。

  自定义事件用于收集任意您期望跟踪的数据。

二、 接入数据系统

1. 为应用申请 APP ID

在 analytics.talkingdata.net 网站中创建一款应用，您将获得一串 32 位的 16 进制 APP ID，用于唯一标识您的一款应用。TalkingData 支持跨平台的应用，跨平台应用只需获得一个 APP ID 即可。

2. 向工程中导入追踪 SDK

下载数据统计 SDK 后解压至本地目录，将其中的*.h 和 lib*.a 导入到您的应用工程中；

在您的工程里→选择 File --> Add Files to"Your Project"--> 选择 TalkingData.h 和 libTalkingData.a 两个文件勾选 Copy items into destination group's folder (if needed) 并确保所有要用到 SDK 的 targets 都处于选中状态。

3. 添加依赖框架（FRAMEWORK）

TalkingData Analytics 需要使用 Security. framework 来辅助存储设备标识，CoreTelephony.framework 框架获取运营商标识，使用 AdSupport.framework 获取 advertisingIdentifier，使用 libz.dylib 进行数据压缩。 Xcode 的添加方式如下所示：

xcode 版本	操作
xcode5.x	在您的工程里，选择 target-->Build Phases-->Link Binary With Libraries，点击 + 号，选择 Security.framework、CoreTelephony.framework、AdSupport.framework 和 libz.dylib

如要支持6.0以下系统，请将AdSupport.framework依赖条件改成Optional，如下图：

4. 添加必须的调用方法（重要）

使用数据统计系统需要至少添加以下调用方法，这些调用用于准确跟踪用户每次的应用使用，是准确统计启动、活跃、留存数据的基础：

- 在需要调用 SDK 方法的文件中导入 TalkingData.h；
- 在 application:didFinishLaunchingWithOptions:方法里面调用[TalkingData sessionStarted:@"您的 AppKey" withChannelId:@"channel_id（可选参数）"]。

  附注：已创建完成的应用，可进入对应应用的数据报表页中，**在“系统设置”-“编辑应用”页面里能随时查看到获得的 APP ID。**
- channel_id 是渠道标识符，可通过为不同越狱推广渠道分配不同的渠道标识安装包的方式来单独追踪数据，AppStore 中提交的版本亦可单独进行标记。不需要区分渠道时，可以传空字符串“”。但请注意，每台设备的全部数据会计入首个安装激活的渠道，用户更换渠道包后不重复计算新增。

示例：

```
#import "TalkingData.h"

- (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions{
      [TalkingData sessionStarted:@"ae878999xxxxxx"
                      withChannelId:@"weiphone"];
      //other code

}
```

三、 更多高级功能

1. 追踪应用的页面使用情况

此功能可帮助开发者统计应用中各个页面的访问次数和停留时长，为产品优化提供依据。您需要在要追踪的页面被打开和关闭时调用以下方法，来获取数据：

- 在 viewWillAppear 或 viewDidAppear 方法里调用 trackPageBegin 方法：[TalkingData trackPageBegin:@"page_name"];
- 在 viewWillDisappear 或者 viewDidDisappear 方法里调用 trackPageEnd 方法：[TalkingData trackPageEnd:@"page_name"];

- 其中 page_name 是您自定义的页面名称。

  格式：支持 64 个字符以内的英文、数字、下划线的混合名称。请一定注意不要加入空格或其他开发中的转义字符。

2. 纠正用户的地区信息

TalkingData 默认使用设备中收取的 MCC（移动国家码）和用户联网 IP 来判定用户的地区，与地区相关的数据会有一定误差。

如果您的应用会使用用户的位置信息，可通过接口将信息提交至 TalkingData 数据中，可使您获得更加精准的数据报表。调用以下方法来提交您获取的精确信息：

```
[TalkingData setLatitude:纬度  longitude:经度];
```

3. 使用自定义事件

使用说明

自定义事件用于统计任何您期望去跟踪的数据，如：点击某功能按钮、填写某个输入框、触发了某个广告等；同时，自定义事件还支持添加一些描述性的属性参数，可使用 Key-Value 的方式来进行发送（非必须使用），对某一次事件可传送多达 10 个参数。

一些事件会具有层级关系，如果给每个都命名不同的 Event ID，会使 ID 总数量过多，带来管理和查阅数据的麻烦，我们提供了 Label 标签的用法。如果一系列事件都属于同一类，可使用相同的 Event ID，而选用不同的 Label 即可。这个用法类似于 Event ID 是大目录，Label 是具体事件内容。

- Event ID 无需提前在数据平台中定义，可自行定义名称，直接加入到应用中需要跟踪的位置即可生效。TalkingData 最多支持 100 个不同 Event ID。如果您要跟踪的事件过多，请做好分类，善用 Label 即可达到您的目的。
- 格式：32 个字符以内的中文、英文、数字、下划线，注意 eventId 中不要加空格或其他的转义字符。

调用方法

- 在应用程序要跟踪的事件处加入下面格式的代码，也就成功的添加了一个简单事件到您的应用程序中了：

  ```
  [TalkingData trackEvent:@"event_id"];
  ```

- 跟踪多个同类型事件，无需定义多个 Event ID，可以使用 Event ID 做为目录名，而使用 Label 标签来区分这些事件，可按照下面格式添加代码：

  [TalkingData trackEvent:@"event_id" label:@"event_label"];

- 为事件添加详尽的描述信息，可以更有效的对事件触发的条件和场景做分析。您可以使用 NSDictionary 对象上传事件参数，key 类型必须是 NSString，value 可以是 NSString 戒者 NSNumber，一次事件最多只支持 10 个参数：

  [TalkingData trackEvent:@"event_id" label:@ "label"
  parameters:Your_dictionary];

- 如果所有事件都需要传输相同的参数，可以设置全局的 Key-Value，这些 Key-Value 会自动添加到所有自定义事件：

  [TalkingData setGlobalKV :key value:value];

  如果要移除全局 Key-Value，则可以调用：

  【TalkingData removeGlobalKV: key];

  注：如果 onEvent 里传入的 Key-Value 里的 key 和全局 Key-Value 里的 key 冲突，以 onEvent 里传入的为准。

示例 1：

跟踪某电商应用中首页的 5 个不同推广位置的点击次数，并收集宣传品的品类、促销价格范围等信息：

```
// 可定义 eventId=点击首页推荐位；event_LABEL=具体的位置编号
#define recommandClick @"首页推荐位点击"
NSDictionary *dic = [ [NSDictionary alloc] init];
[dic setObject:@"服装" forKey:@"商品类别"];
[dic setObject:@"5~10" forKey:@"price"];
[TalkingData trackEvent: recommandClick label: @"第一广告位"
parameters:dic];
[dic release]
NSDictionary *dic2 = [ [NSDictionary alloc] init];
[dic setObject:@"家电" forKey:@"商品类别"];
[dic2 setObject:@"500~1000" forKey:@"price"];
[TalkingData trackEvent: recommandClick label: @"第三广告位" parameters
```

```
dic2];
[dic2 release]
```

注 1 :这里需追踪内容属于同类型 ,使用了 Label 标签的用法。以相同 Event ID 来作为统一名称，可收集到 5 个位置的总量点击数据，Label 来区分不同的具体位置，可单独查看不同位置的数据。

注 2：在 value 取值较离散情况下（如示例中收集的促销价格信息），不要直接填充具体数值，而应划分区间后传入（如 9.9 元，可定义 5～10 元的价格区间，传入 5～10 元），否则 value 不同取值很可能超过平台最大数目限制，而影响最终展示数据的效果。

示例 2：

在一款休闲游戏中记录玩家在各关卡中的失败数据，并收集玩家具体信息。

```
// 可定义 eventId=战斗失败
NSDictionary *dic = [ [NSDictionary alloc] init];
[dic setObject:@"20-30" forKey:@"等级"];//级别区间
[dic setObject:@"沼泽地阿卡村" forKey:@"关卡名"]; //关卡名称
[dic setObject:@"主动退出" forKey:@"失败原因"]; //失败原因
[dic setObject:@"10000～12000" forKey:@"coin"]; //携带金币数量
[TalkingData trackEvent:@"战斗失败" label:nil parameters:dic];
[dic release];
```

4. 收集应用错误日志

收集应用的错误日志可帮助您来修正 BUG，改善产品。为简化开发者的工作，我们提供自动获取异常信息的功能 ,并会记录发生事件、原因、堆栈调用情况等信息。但考虑异常信息收集会损耗最终用户流量，因此自动捕获默认为关闭状态，您可根据需要开启此功能。

- 可向以下方法中传入 YES（开启自动捕获）或 NO（关闭自动捕获），来控制自动捕获功能的开关：

  [TalkingData setExceptionReportEnabled:NO];

- 注意，以上调用方法必须早于“在 application:didFinishLaunchingWithOptions: 方法里面调用 sessionStarted”之前调用，否则调用可能会生效。

四、 重点问题解答

问题 1：TalkingData 是否支持 iOS OS 5.0？

解答：我们编译 SDK 时的设置为 Base SDK：iOS 7.0, iOS Deployment Target：iOS 5.1.1，代码里面带有版本判断，可兼容 iOS OS 5.1.1 以上版本。

问题 2：TalkingData 如何确定唯一设备，是否用了 UDID？

解答：未调用。TalkingData SDK 正式版自推出时起从未获取设备的 UDID。当前 SDK 使用以 IDFA 为基础的信息来区别唯一设备，同时，TalkingData 在设备本地存储了我们生成的随机唯一 ID，以避免任何苹果的政策改动带来的风险。