# 奥丁数据 IOS数据收集SDK用户手册

|  |  |  |  |
| --- | --- | --- | --- |
| 项目代号 | 文档类型 | 发布版本 | 发布日期 |
|  | Doc | V1.0 |  |
| 编制 | 丁次 | 编制日期 | 2019/05/12 |
| 审核 |  | 审核日期 |  |
| 批准 |  | 批准日期 |  |

文件修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 版本 | 修订人 | 修订项 | 修订时间 |
| 1 | V1.0 | 丁次 | 功能优化 | 2019/06/06 |
| 2 | V1.0.0\_beta | 丁次 | 兼容到iOS8，代码功能优化等 | 2019/06/15 |
| 3 | 1.0.62 | 丁次 | 游客事件优化 | 2019/10/11 |
| 4 | 1.0.63 | 丁次 | 时间国际化 | 2019/11/09 |
| 5 | 1.0.64 | 丁次 | 通用事件，新增设备属性 | 2019/11/11 |
| 6 | 1.0.7 | 丁次 | 域名安全 | 2020/01/03 |
| 7 | 1.0.8 | 丁次 | 版本优化 | 2020/01/19 |

[奥丁数据 IOS数据收集SDK用户手册 1](#_Toc29027523)

[1.产品介绍 4](#_Toc29027524)

[2.功能概述 4](#_Toc29027525)

[3.功能详述 4](#_Toc29027526)

[3.1集成准备 4](#_Toc29027527)

[3.1.1设置ShareSDK的appkey 4](#_Toc29027528)

[3.2 手动集成 4](#_Toc29027529)

[3.3 接口功能介绍 5](#_Toc29027530)

[3.3.1初始化接口 5](#_Toc29027531)

[3.3.2 页面统计 5](#_Toc29027532)

[3.3.3 用户注册 5](#_Toc29027533)

[3.3.4 用户登录 6](#_Toc29027534)

[3.3.5 支付事件 7](#_Toc29027535)

[3.3.6 操作事件 7](#_Toc29027536)

## 1.产品介绍

数据收集SDK，用于获取用户及设备等基本属性，提供常用事件接口及自定义事件接口，自动页面追踪及崩溃捕捉，分析用户行为及数据统计等;

## 2.功能概述

数据采集SDK，将用户在APP应用中产生的行为数据，进行采集并上报到日志API服务器。日志API服务器对用户行为数据进行接收并响应，再将数据进行转码、解压后生产到

消息队列Kafka中。大数据处理程序，会每隔5秒钟去消费消息队列中的用户行为数据，并根据一定的过滤规则，将行为数据进行清洗、整理并入库。

## 3.功能详述

### 3.1集成准备

AppKey是奥丁数据平台+后台用来标示App的唯一标识符，集成SDK前需要在创建应用并获取相应的AppKey。请开发者到奥丁数据平台注册自己的账号，创建运用程序并获取对应的AppKey.

#### 3.1.1设置ShareSDK的appkey

**在项目工程的Info.plist 中增加OidnKey 和 OdinSecret 两个字段**

填入平台获取的odinKey和odin**Secret的值**

### 3.2 手动集成

① 依赖库：

CoreTelephony.framework 获取运营商标识

libz.tbd 数据压缩

libsqlite.tbd 数据缓存

SystemConfiguration.framework 判断网络状态

libsqlite3.tbd 数据存储

② 工程配置

选择SDK并下载，解压.zip文件得到相应的组件包

XcodeFile —> Add Files to "Your Project"，在弹出Panel选中所下载组件包－>Add。（注：选中“Copy items if needed”）添加依赖库，在项目设置target -> 选项卡General ->Linked Frameworks and Libraries

Build Settings-> other linker Flags 添加-ObjC

### 3.3 接口功能介绍

#### 3.3.1初始化接口

初始化方法必须在didFinishLaunchingWithOptions中调用

/\*

\* 初始化SDK

\* @param appkey APP注册时获取唯一标识符

\* @param channel 当前使用渠道，设置nil默认APP Store

\*/

+ (void)initSDKWithAPPKey:(NSString \*)appkey Channel:(NSString \*)channel;

#### 3.3.2 页面统计

UIViewController 根据生命周期自动监控

针对UIPageViewController，请在子控制的viewWillAppear:和viewWillDisappear:中配对调用/\*\*

针对PageView的统计

@param pageName 开始页面的名称

\*/

+ (void)pageStart:(NSString \*)pageName;

/\*\*

针对PageView的统计

@param pageName 结束页面的名称

\*/

+ (void)pageEnd:(NSString \*)pageName;

#### 3.3.3 用户注册

记录宿主app用户注册事件注册 开发者根据自身代码情况主动埋点收集 app用户注册事件

/\*

\* 注册事件统计

\* @param userId 用户id(用户注册成功后获取)

\*/

+ (void)eventRegister:(NSString \*)userId;

#### 3.3.4 用户登录

① 单纯记录宿主app用户登录结果，开发者主动埋点收集

/\*

\* 登录接口

\* @param userId 用户id(用户登录成功后获取)

\* @param state 登录状态;1成功0失败。

\*/

+ (void)eventLogin:(NSString \*)userId status:(NSInteger)status;

② 记录宿主app用户登录事件过程记录，登录动作开始到结束的耗时结束的一个记录收集

一次成对的 beginEventLogin -> endEventLogin调用，当前版本sdk默认宿主app全局只有一个登录入口 ，开发者主动埋点收集

登录动作触发时做登录开始动作埋点使用beginEventLogin方法

/\*

\* 开始登录统计（用于统计登录时长）

\*/

+ (void)beginEventLogin;

登录结束得到服务器的反馈此时使用endEventLogin结束整个登录事件数据收集

/\*

\* 结束登录统计(配合开始登录使用)

\* @param userId 用户id(用户登录成功后获取，失败传空)

\* @param state 登录状态;1成功0失败。

\*/

+ (void)endEventLogin:(NSString \*)userId status:(NSInteger)status;

#### 3.3.5 支付事件

支付事件是当前版本SDK 默认设置的一种事件类型用于记录一单交易的发生，一单交易成功后方可进行事件的收集 。开发者主动埋点收集

/\*

\* 支付事件

\* @param payAmount 支付金额

\* @param orderNumber 订单编号

\* @param amount 数量

\*/

+ (void)eventPay:(double)payAmount orderNumber:(NSString \*)orderNumber amount:(NSString \*)amount;

#### 3.3.6 操作事件

收集用户在宿主app中的某一控件上做的交互事件（点击、滑动、等）。开发者主动埋点收集

+ (void)eventOperation:(NSString \*)eventId;

例子：页面刷新按钮点击, 如下单纯记录一次 刷新按钮点击

[AoDAnalysisSDK eventOperation:@"refresh\_oclick"];

SDK自动捕获宿主app 软件运行异常宿主app内部业务异常（登录失败异常，支付异常，等）收集 ，开发者根据自身业务需求进行埋点

/\*\*

异常事件统计

@param title 异常标题

@param context 异常内容

\*/

+ (void)eventErrorTitle:(NSString \*)title context:(NSString \*)context;

Content,String errorTitle)