

• [首页](#) • [开源项目](#) • [问答](#) • [代码](#) • [博客](#) • [翻译](#) • [资讯](#) • [移动开发](#) • [招聘](#) • [城市圈](#)
当前访客身份：游客 [[登录](#) | [加入开源中国](#)]

在 40196 款开源软件中搜

软件 ▼

软件

搜索



[xrzs](#) [关注此人](#)

[关注\(2\)](#) [粉丝\(941\)](#) [积分\(429\)](#)

1) Writing is thinking. 2) If you're interesting someone won't agree with what you said.

[.发送私信](#) [.请教问题](#)

博客分类

- [java](#)(43)
- [javame/android](#)(1)
- [python](#)(57)
- [perl](#)(2)
- [scala](#)(1)
- [awk/shell](#)(24)
- [sqlite](#)(1)
- [sql技巧](#)(6)
- [mysql](#)(98)
- [oracle](#)(4)
- [tcp/ip/http](#)(11)
- [JavaNet/Multithread](#)(5)
- [network](#)(16)
- [linux/unix](#)(38)
- [windows/office](#)(5)
- [php/nginx/apache](#)(14)
- [jsp/mvc/ssh](#)(3)
- [wsgi/Flask](#)(7)
- [js/ajax/extjs/jquery](#)(12)
- [hbase](#)(12)
- [pig](#)(6)
- [NoSQL](#)(12)
- [hadoop](#)(39)
- [hadoop生态圈](#)(3)
- [spark](#)(5)
- [hive](#)(30)
- [BI/DW/DP](#)(5)
- [DA/DM/ML](#)(13)
- [algorithm](#)(7)
- [data analysis theory](#) (27)
- [日志采集与架构](#)(5)
- [职场·杂谈](#)(38)
- [管理之道](#)(7)

- [工作笔记](#)(19)
- [resource](#)(1)
- [QA](#)(2)
- [金融·理财](#)(1)

阅读排行

1. [1. 关于 python ImportError: No module named 的问题](#)
2. [2. 循序渐进Java Socket网络编程 \(多客户端、信息共享、文件传输 \)](#)
3. [3. Flume NG 简介及配置实战](#)
4. [4. Eclipse 下找不到或无法加载主类的解决办法](#)
5. [5. 动态规划算法之：最长公共子序列 & 最长公共子串 \(LCS \)](#)
6. [6. 关于 HTTP GET/POST 请求参数长度最大值的一个理解误区](#)
7. [7. storm 原理简介及单机版安装指南](#)
8. [8. python 日志模块 logging 详解](#)

最新评论

- [@河边沉思](#)：学习了 [查看»](#)
- [@赛克蓝德](#)：可以直接用工具Piwik等 [查看»](#)
- [@PREPOET](#)：后面部分完全一样 [查看»](#)
- [@素人派surenpi](#)：学习了 [http://surenpi.com](#) [查看»](#)
- [@Json_xu](#)：枚举的不错 [查看»](#)
- [@just4scala](#)：写的真好 [查看»](#)
- [@kyle_wang](#)：[查看»](#)
- [@lyhabc](#)：rsync报错protocol version mismatch is your s... [查看»](#)
- [@Leon_wy](#)：2 [查看»](#)
- [@进击的柯南](#)：整理的很系统，多谢 [查看»](#)

访客统计

- 今日访问：715
- 昨日访问：1534
- 本周访问：5277
- 本月访问：3783
- 所有访问：742868

[空间](#) » [博客](#) » [日志采集与架构](#)

转 iOS 统计打点那些事

发表于6个月前(2015-09-11 00:40) 阅读 (607) | 评论 (0) 10人收藏此文章, [我要收藏](#)

赞0



DDoS高防IP 300G无限防

马上抢购

目录[-]

- [1、统计代码埋点真的 so easy ?](#)
- [2、AOP 实现业务代码与统计代码解耦](#)
- [3、多个 Button 事件怎么办 ?](#)

- [4、统计事件有选择逻辑呢？](#)
- [5、如何避免统计代码被遗忘？](#)
- [6、如何解决 APP 埋码要发版的问题？](#)
- [7、Refer：](#)

1、统计代码埋点真的 so easy？

统计打点是 App 开发里很重要的一个环节，App 的运行状态、改版后的效果、用户的各种行为等都需要打点，市面上也有不少可供选择的第三方库。假设产品有这么个需求：当用户在详情页点击购买按钮时，记录一下事件。我们实现起来大概会是这样

```
1 // DetailViewController.m
2
3 - (void)onBuyButtonTapped:(UIButton *)button
4 {
5     // do some stuff, maybe send a request to server
6     [XXXAnalytics event:kSomeEventYouDefined];
7 }
```

这个需求就这样轻松搞定了，但细细想想还是有不少问题的：

- 页面上会有其他的 Button，可能每个 Button 都要放上这么一段代码。
- 这些统计其实跟具体的业务无关，没必要跟业务代码混杂在一起，不优雅。
- 当改版或者重构时，有可能忘了把相应的打点代码迁移过去。

2、AOP 实现业务代码与统计代码解耦

所以需要一种更好的方式来做这件事，这就是使用 AOP(Aspect-Oriented-Programming)，翻译过来就是「面向切面编程」

通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。

简单来说，就是可以动态的在函数调用的前后插一段代码。iOS 可以使用 Pete Steinberger 开发的 Aspects 这个库，大致原理是在 runtime 层，通过 swizzle method 来实现的。

来看一个小 Demo

```
1 [UIViewController aspect_hookSelector:@selector(viewWillAppear:) withOptions?
2     NSLog(@"View Controller %@ will appear animated: %tu", aspectInfo.instance
3     } error:NULL];
```

这样在 UIViewController 的 viewWillAppear: 被调用后，还会再调一下我们定义的 Block，这段日志就会被输出。而打点正好符合这种场景：正事干完之后，额外干一些跟业务无关的事情。

上面的例子，我们通过 AOP 来做的话，大概就是这样

```
1 // DetailViewController.m
2 - (void)onBuyButtonTapped:(UIButton *)button
3 {
4     // do some stuff, maybe send a request to server
5     // no need to call [XXXAnalytics event:]
6 }
```

```
7
8 // AppDelegate.m
9 - (void)setupAnalytics
10 {
11     [DetailViewController aspect_hookSelector:@selector(onBuyButtonTapped)
12     [XXXAnalytics event:kSomeEventYouDefined];
13     } error:NULL];
14 }
```

3、多个 Button 事件怎么办？

这样统计代码就从业务代码中剥离出来了。但是又产生了一个新问题，多个 Button Event，岂不是要写很多行这样的代码，「重复」这样的事情，作为一个程序员怎么能忍，简单，造一个方法

```
1 - (void)trackEventWithClass:(Class)klass selector:(SEL)selector event:(NSString*)event
2 {
3     [klass aspect_hookSelector:@selector(selector) withOptions:AspectPositionBefore
4     [XXXAnalytics event:event];
5     } error:NULL];
6 }
```

使用起来就像这样

```
1 - (void)setupAnalytics
2 {
3     [self trackEventWithClass:DetailViewController selector:@selector(onBuyButtonTapped) event:kSomeEventYouDefined];
4     [self trackEventWithClass:ListViewController selector:@selector(followButtonTapped) event:kSomeEventYouDefined];
5     // ...
6 }
```

4、统计事件有选择逻辑呢？

看起来又干净了些。这时，产品经理又提了个需求：当这个按钮点击时，如果已经登录了，发送 EventA，如果没有登录则发送 EventB，也就是说，不再只是 [XXXAnalytics event:] 这么简单了，还需要加上额外的逻辑，这也难不倒我们，加上一个 block 即可。

```
1 - (void)trackEventWithClass:(Class)klass
2     selector:(SEL)selector
3     eventHandler:(void (^)(id<AspectInfo> aspectInfo))eventHandler
4 {
5     [klass aspect_hookSelector:@selector(selector) withOptions:AspectPositionBefore
6     if (eventHandler) {
7         eventHandler(aspectInfo);
8     }
9     } error:NULL];
10 }
11
12 // 使用
13 [self trackEventWithClass:DetailViewController selector:@selector(onBuyButtonTapped)
14     user.loggedIn ? [XXXAnalytics event:EventA] : [XXXAnalytics event:EventB]
15     ]];
```

好了，现在只要不是太复杂的打点逻辑(那些需要方法上下文变量的)我们都能应付了，接下来就该等产品来验收了。产品搬了个凳子坐在身边，然后点一下 Button，看一下 Console，被几轮蹂躏后，产品也慢慢地接受了这种验收方式。后来某一天，忽然发现某一项或某几项数据有异常，然后找到开发，瞄

5、如何避免统计代码被遗忘？

那么有没有一种直观的验证方式呢？当然，程序员是万能的呀。一个理想的状况是，产品打开 App 后，开启某个开关就能看到所有会发送 Event 的按钮，就像这样



#脸大才能干遮面# 自拍瘦脸不一一遮脸是力能
哒，托下巴那一挡，歪头让头发一盖，五官里随
便哪个一遮，拿个物体入镜什么的都是满满的心
机呀～歪头也是...



逛逛逛



买买买



聊聊聊



我我我

其中数字代表了 EventID。如何实现呢？还记得注册事件时，我们有传入 class 和 selector 么，一般我们都会有一个 BaseViewController，那么就可以在 BaseViewController 的 viewDidLoad: 里做点文章了。

```
1 // BaseViewController.m
2
3 - (void)viewDidLoad:(BOOL)animated
4 {
5     [super viewDidLoad:animated];
6     // 获取已经注册过的 classes
7     NSDictionary *registeredClasses = [OurAnalytics sharedInstance].register
8
9     [registeredClasses enumerateKeysAndObjectsUsingBlock:^(NSString *className,
10         if ([self isKindOfClass:NSClassFromString(className)]) {
11             // 如何根据 selector 找到它的宿主?
12         }
13     }];
14 }
```

所以现在问题就剩下，如何根据 selector 找到对应的 Button，这里要注意，有些 Button 可能要等网络请求完成才会出现，比如 TableViewCell 里的 Button。

没有想到太方便的方法，简单粗暴点就是设置个 Timer 每隔一段时间扫一下 subviews，如果是 button 或 包含 tapGesture 的，就拿它们的 action 对比一下，如果 match 就可以高亮那个 button / view 了。

EventID 也一样，之前在注册时也会传一个 EventID 过来，这里直接显示出来即可。对于那些传 eventHandler 的就不行了。

所以理论上是可行的，性能上会稍微有点损耗，尤其是当 subViews 的结构比较复杂时，不过只是内部用来做验证，所以这也不是什么问题。

6、如何解决 APP 埋码要发版的问题？

看起来效果已经不错了，有没有可能让这套体系再灵活一些？比如可以从后端制定打点规则？客户端只是读取一个配置文件，就像这样

```
1 - (void)setupAnalytics
2 {
3     // analyticsRules 是从配置文件中读取出来的
4     [analyticsRules enumerateObjectsUsingBlock:^(NSDictionary *rules, NSUInteger index, BOOL *stop, SEL selector) {
5         Class klass = NSClassFromString(rules[@"class"]);
6         SEL selector = NSSelectorFromString(rules[@"selector"]);
```

```

7 | NSString *eventID = rules[@"eventID"];
8 | [self trackEventWithClass:klass selector:selector event: eventID];
9 |     }
10| }

```

那如果在后台的时候填错了 Class 或 Selector 怎么办？还好有 `objc_getClassList` 和 `class_copyMethodList` 这两个运行时方法，有了它们就可以在 App 启动时扫一遍已注册的类（过滤掉 UI / NS 开头的），然后将它们的 selector 也一并保存下来发送给服务端，当然这种操作只需在适当的时机做一下就可以了，比如集成打包时。

现在，这套体系就比较完整了。当然这只是我的一些构想，并没有在实践中尝试过，所以肯定会踩到各种各样的坑，不过至少看起来是个可行的方案。

7、Refer：

[1] iOS 统计打点那些事

<http://limboy.me/ios/2015/09/09/ios-analytics.html>

[2] 基于Aspects和JSPatch的热埋点方案

<http://t.cn/RyGQ9wm>

[3] 移动APP日志上报优化实践

<http://dwz.cn/24BfTW>

分享到： 新浪微博  腾讯微博  0赞

原文地址：<http://limboy.me/ios/2015/09/09/ios-analytics.html>

- [« 上一篇](#)
- [下一篇 »](#)



PARIS ↔ LONDRES
*conditions sur ouibus.fr



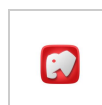
Trouvez un instant de bien-être et de beauté
www.balineia.com



Tout pour vos animaux de compagnie
morinfrance.com

最新热门职位

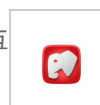
更多开发者职位上 [开源中国·招聘](#)



高级Java研发工程师 趣互
联

月薪：10-20K
高级服务器开发工程师
趣互联

月薪：10-20K



高级J2EE研发工程师 趣互
联

月薪：10-20K
电商产品经理 广西云天实
业有限公司

月薪：18-25K

评论0



插入：[表情](#) [开源软件](#)

发表评论

[关闭](#)插入表情

关闭相关文章阅读

- 2013/01/08 [关于页面来源统计的一点点思考](#)
- 2014/10/17 [图说ios app数据](#)
- 2013/11/05 [MySQL 按时间统计](#)
- 2012/11/11 [wordpress添加网站统计](#)
- 2014/04/22 [\[转\] SQL统计](#)

© 开源中国(OSChina.NET) | [关于我们](#) | [广告联系](#) | [@新浪微博](#) | [开源](#) 开源中国手机客户端：
[中国手机版](#) | 粤ICP备12009483号-3

开源中国社区(OSChina.net)是工信部 [开源软件推进联盟](#) 指定的官方社区