链滴

iOS 开发之 Method Swizzling 深入浅出

作者：liman

原文链接：https://ld246.com/article/1524562646998

来源网站：链滴

=======

只要善用Google，网上有很多关于Method Swizzling的Demo，在这里我就不打算贴代码了，主要绍下概念，原理，注意事项等等。

开发需求

如果产品经理突然说:"在所有页面添加统计功能，也就是用户进入这个页面就统计一次"。我们会想到面的一些方法：

- 手动添加

直接简单粗暴的在每个控制器中加入统计，复制、粘贴、复制、粘贴...

上面这种方法太Low了，消耗时间而且以后非常难以维护，会让后面的开发人员骂死的。

- 继承

我们可以使用继承的方式来解决这个问题。创建一个基类，在这个基类中添加统计方法，其他类都继自这个基类。

然而，这种方式修改还是很大，而且定制性很差。以后有新人加入之后，都要嘱咐其继承自这个基类所以这种方式并不可取。

- Category

我们可以为UIViewController建一个Category，然后在所有控制器中引入这个Category。当然我们可以添加一个PCH文件，然后将这个Category添加到PCH文件中。

- Method Swizzling

我们可以使用苹果的“黑魔法”Method Swizzling，Method Swizzling本质上就是对IMP和SEL进交换。

先了解几个概念

Selectors, Methods, & Implementations

在 Objective-C 的运行时中，selectors, methods, implementations 指代了不同概念，然而我们通会说在消息发送过程中，这三个概念是可以相互转换的。 下面是苹果 Objective-C Runtime Referen e中的描述：

- Selector（typedef struct objc_selector *SEL）:在运行时 Selectors 用来代表一个方法的名字。Se ector 是一个在运行时被注册（或映射）的C类型字符串。Selector由编译器产生并且在当类被加载进存时由运行时自动进行名字和实现的映射。
- Method（typedef struct objc_method *Method）:方法是一个不透明的用来代表一个方法的定的类型。

- Implementation（typedef id (*IMP)(id, SEL,...)）:这个数据类型指向一个方法的实现的最开始的方。该方法为当前CPU架构使用标准的C方法调用来实现。该方法的第一个参数指向调用方法的自身即内存中类的实例对象，若是调用类方法，该指针则是指向元类对象(metaclass）。第二个参数是这方法的名字selector，该方法的真正参数紧随其后。

理解 selector, method, implementation 这三个概念之间关系的最好方式是：在运行时，类（Class维护了一个消息分发列表来解决消息的正确发送。每一个消息列表的入口是一个方法（Method），个方法映射了一对键值对，其中键值是这个方法的名字 selector（SEL），值是指向这个方法实现的数指针 implementation（IMP）。 Method swizzling 修改了类的消息分发列表使得已经存在的 selctor 映射了另一个实现 implementation，同时重命名了原生方法的实现为一个新的 selector。

Method Swizzling原理

Method Swizzing是发生在运行时的，主要用于在运行时将两个Method进行交换，我们可以将Method Swizzling代码写到任何地方，但是只有在这段Method Swilzzling代码执行完毕之后互换才起作用。

Method Swizzling 使用注意

类簇设计模式

在iOS中NSNumber、NSArray、NSDictionary等这些类都是类簇(Class Clusters)，一个NSArray的现可能由多个类组成。

所以如果想对NSArray进行Swizzling，必须获取到其“真身”进行Swizzling，直接对NSArray进行作是无效的。

下面列举了NSArray和NSDictionary本类的类名，可以通过Runtime函数取出本类。

类名	真身
NSArray	__NSArrayI
NSMutableArray	__NSArrayM
NSDictionary	__NSDictionaryI
NSMutableDictionary	__NSDictionaryM

注意要点

- Swizzling应该总在 +load中执行
- Swizzling应该总是在 dispatch_once中执行
- Swizzling在 +load中执行时，不要调用[super load]。如果多次调用了[super load]，可能会出“Swizzle无效”的假象，原理见下图：

Swift 自定义类中使用 Method Swizzling

要在 Swift 自定义类中使用 Method Swizzling 有两个必要条件：

- 包含 Swizzle 方法的类需要继承自 NSObject

原文链接：iOS 开发之 Method Swizzling 深入浅出

- 需要 Swizzle 的方法必须有动态属性（dynamic attribute）

注：对于 Swift 的自定义类，因为默认并没有使用 Objective-C 运行时，因此也没有动态派发的方法表，所以如果要 Swizzle 的是 Swift 类型的方法的话，是需要将原方法和替换方法都加上 dynamic 记，以指明它们需要使用动态派发机制。当然类也要继承自 NSObject。

再注：下面这个例子使用了 Objective-C 的动态派发，对于 NSObject 的子类（UIViewController 是可以直接使用的，并不是 Swift 中自定义的类，因此没有加 dynamic 标记也是可以的。

Method Swizzling 中 Objective-C 与 Swift 的异同

区别	Objective-C	Swift
Runtime 头文件需要	#import <objc/runtime.h>	
Swizzling 调用处	load 方法	initialize 方法

注：load 方法只在 Objective-C 里有，而且不能在 Swift 里重载，不管怎么试都会报编译错误。接来执行 Swizzle 最好的地方就是 initialize 了，这是调用第一个方法前的地方。

因为 Swizzling 会改变全局状态，所以我们需要在运行时采取一些预防措施。GCD 的dispatch_once 可以保证操作的原子性，确保代码只被执行一次，不管有多少个线程。

Method Swizzling 实际应用

APM(应用性能管理)

网络监控的原理，应该就是hook NSURLConnection，NSURLSession 。崩溃收集的原理，应该就是ook NSException。

- https://newrelic.com/ 国外行业老大
- http://www.tingyun.com/ 国内听云
- http://www.oneapm.com/ 国内OneAPM
- http://apm.netease.com/ 国内网易

国外资料 :nerd:

可能需要梯子

- http://nshipster.com/method-swizzling/
- https://medium.com/rocknnull/ios-to-swizzle-or-not-to-swizzle-f8b0ed4a1ce6
- https://medium.com/@abhimuralidharan/method-swizzling-in-ios-swift-1f38edaf984f
- https://darkdust.net/writings/objective-c/method-swizzling
- https://blog.newrelic.com/2014/04/16/right-way-to-swizzle/
- https://spin.atomicobject.com/2014/12/30/method-swizzling-objective-c/
- https://wiredcraft.com/blog/method-swizzling-ios/

- https://www.raywenderlich.com/177890/swizzling-in-ios-11-with-uidebugginginformationoerlay
- http://petersteinberger.com/blog/2014/a-story-about-swizzling-the-right-way-and-touch-frwarding/
- https://iossolves.blogspot.com/2017/11/swift-4-method-swizzling-part-12.html
- http://iossolves.blogspot.com/2017/11/swift-4-method-swizzling-part-22.html
- https://sexyswift.wordpress.com/tag/method-swizzling/
- https://qiita.com/paming/items/25eaf89e4f448ab05752
- http://www.swift-studies.com/blog/2014/7/13/method-swizzling-in-swift
- https://academy.realm.io/posts/sash-zats-swift-swizzling/

w □

GitHub开源了一款iOS调试小工具，功能之一就是实现网络请求抓包（代码零入侵）,原理也是使用了Method Swizzling, 感兴趣的童鞋可以进来看看, 也欢迎使用□ http://DotzuX.com