

学习RAC小记-适合给新手看的RAC用法总结



作者 cbsfly_iDev (/u/4daf2d76e124) [+ 关注](#)

2016.01.04 21:06* 字数 1855 阅读 23720 评论 28 喜欢 136
(/u/4daf2d76e124)

最近叶大直播写代码，我也做点小笔记。

什么是RAC?

几乎每一篇介绍RAC的文章开头都是这么一个问题。我这篇文章是写给新手（包括我自己）看的，所以这个问题更是无法忽视。

简单的说，RAC就是一个第三方库，他可以大大简化你的代码过程。

官方的说，ReactiveCocoa (<https://github.com/ReactiveCocoa/ReactiveCocoa>)（其简称为RAC）是由GitHub开源的一个应用于iOS和OS X开发的新框架。RAC具有函数式编程 (<https://zh.wikipedia.org/wiki/%E5%87%BD%E6%95%B8%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80>)和响应式编程 (<https://zh.wikipedia.org/wiki/%E5%93%8D%E5%BA%94%E5%BC%8F%E7%BC%96%E7%A8%8B>)的特性。

为什么我们要学习RAC?

为了提高我们的开发效率。RAC在某些特定情况下开发时可以大大简化代码，并且目前来看安全可靠。

配置RAC环境

我习惯用cocoapods来安装github上得开源库，不会的新手iOS开发者有兴趣可以去学一下。

想学习cocoapods的同学推荐唐巧前辈的文章 (<http://devtang.com/blog/2014/05/25/use-cocoapod-to-manage-ios-lib-dependency/>)。

```
platform:ios, '8.0'  
pod 'ReactiveCocoa', '~>2.1.8'
```

这里有一点要注意下就是RAC的版本问题，由于还没学习Swift，所以我是用OC编写程序的，最新版的RAC已经支持Swift了，但是在OC的程序安装最新版的RAC可能跑不起来，所以推荐大家使用2.5.0版本以下的RAC（具体支持Swift的版本可能有误，但我引用的2.1.8肯定是没有问题的）。

使用RAC

1.target-action

RAC最基本的入门使用技巧就是对事件的监听。

PS:在iOS开发中, 我们所说的点击事件其实就是target-action, 接触过iOS开发的人都不会陌生UIControlEventTouchUpInside, 这就是按下并松开的动作。不仅仅是UIButton, 还有UITextField也有目标-动作模式。

使用前别忘了引用头文件~

```
#import <ReactiveCocoa/ReactiveCocoa.h>
```

接下来就是最关键的RAC代码了。

```
[[self.textFild rac_signalForControlEvents:UIControlEventEditingChanged] subscribeNext:^(id x){
    NSLog(@"change");
}];
```

就这么短短的两行代码。他实现了一个功能, 即监听了textFild的UIControlEventEditingChanged事件, 当事件发生时实现方法 NSLog 。所以我们可以以这段代码为模板进行RAC的使用, 举一反三, 以后的UIButton点击事件我们都可以用RAC方法进行添加, 再也不用 add Target 了。对于textFild的文字更改监听也有更简单的写法

```
[[self.textFild rac_textSignal] subscribeNext:^(id x) {
    NSLog(@"%@",x);
}];
```

这样就是每次改变TextFild都输出改变后的结果。

再比如给我们的某个label添加一个手势动作, 我们也可以用简单的RAC代码完成

```
UITapGestureRecognizer *tap = [[UITapGestureRecognizer alloc] init];
[[tap rac_gestureSignal] subscribeNext:^(id x) {
    NSLog(@"tap");
}];
[self.view addGestureRecognizer:tap];
```

这段具体我就不解释了, 相信大家都能看得懂, 看不懂的自己写写就懂了。

2.代理

用RAC写代理是有局限的, 它只能实现返回值为void的代理方法

首先我们要明白我们为什么要用RAC写代理? 答: 简化代码! 是的, 的确为了简化代码, 为什么我要再这里强调这个, 是因为在代理方法中我深深的感受到了RAC的优点。一开始我也不愿意花功夫去学RAC, 但是我师父给我举了一个例子, 如果一个View里多个AlertView, 每个AlertView有很多个按钮, 每个按钮都有自己的点击事件, 我应该怎么写? 我想了一下, 不但每个按钮需要打标记, 而且每个AlertView也要打标记, 然后再往代理点击事件里加各种方法, 代码就又臭又长。那么让我们看看RAC怎么写代理方法。

```

UIAlertView *alertView = [[UIAlertView alloc] initWithTitle:@"RAC" message:@"RAC
TEST" delegate:self cancelButtonTitle:@"cancel" otherButtonTitles:@"other", nil];
[[self rac_signalForSelector:@selector(alertView:clickedButtonAtIndex:) fromProto
col:@protocol(UIAlertViewDelegate)] subscribeNext:^(RACTuple *tuple) {
    NSLog(@"%@", tuple.first);
    NSLog(@"%@", tuple.second);
    NSLog(@"%@", tuple.third);
}];
[alertView show];

```

我们来看RAC的语句。@selector 是指这次事件监听的方法 fromProtocol 指依赖的代理。这里block中有一个RACTuple，他相当于是一个集合类，他下面的first，second等就是类的各个参数，我这里点了alertView第二个按钮other输出了一下。

```

2016-01-04 18:24:29.114 RACStudyTest[5003:388870] <UIAlertView: 0x7ff260c90c70; f
rame = (0 0; 0 0); layer = <CALayer: 0x7ff260c91030>>
2016-01-04 18:24:29.115 RACStudyTest[5003:388870] 1
2016-01-04 18:24:29.115 RACStudyTest[5003:388870] (null)

```

可以看出 tuple.second 是ButtonAtIndex中Button的序号。那么对于上面那个我举的例子，就可以用 switch 给各个按钮添加方法，这样的代码看起来更容易理解，方面后期维护。

当然了，alertView代理也有简化的代码。

```

[[alertView rac_buttonClickedSignal] subscribeNext:^(id x) {
    NSLog(@"%@", x);
}];

```

这里的x就是各个Button的序号了，可以直接应对我上述遇到的问题。

3.通知

在我们的开发中通知也是一个比较常用的功能，主要的应用场景是某个页面进行数据重传需要更新model但是点击返回栈时不会刷新返回界面的数据，这时就可以用通知来更新另一个页面的数据，当然我们也可以在另一个页面的 ViewDidAppear 方法中刷新数据，但那是题外话。

这里写的Demo就是我上述说的情况。

首先，在某个页面中我们需要发出通知，这里就是最基本的通知的写法。发送名为 postdata的通知并传送一个数组dataArray。

```

NSMutableArray *dataArray = [[NSMutableArray alloc] initWithObjects:@"1", @"2", @
"3", nil];
[[NSNotificationCenter defaultCenter] postNotificationName:@"postData" object:dat
aArray];

```

而在接受的页面我们需要增加观察者并接受数组，这时我们的RAC就派上用场了。

```

[[NSNotificationCenter defaultCenter] rac_addObserverForName:@"postData" object:
nil] subscribeNext:^(NSNotification *notification) {
    NSLog(@"%@", notification.name);
    NSLog(@"%@", notification.object);
}];

```

当这个页面监听到名为postData的通知时他就会执行block中的方法，当然这里的参数改成 id x 也是可以的，这里用NSNotification主要是强调它的类型。让我们看看控制台的输出。

```
2016-01-04 20:10:52.274 RACStudyTest[5918:439077] postData
2016-01-04 20:10:52.275 RACStudyTest[5918:439077] (
    1,
    2,
    3
)
```

可见，notification.object就是我们想要的数组，当然我们也可以传一些model。值得一提的是，**RAC中的通知不需要 remove observer**，因为在rac_add方法中他已经写了remove。

4.KVO

RAC中得KVO大部分都是宏定义，所以代码异常简洁，简单来说就是 RACObserve(TARGET, KEYPATH) 这种形式，TARGET是监听目标，KEYPATH是要观察的属性值，这里举一个很简单的例子，如果UIScrollView滚动则输出success。

```
UIScrollView *scrollView = [[UIScrollView alloc] initWithFrame:CGRectMake(0, 0, 200, 400)];
scrollView.contentSize = CGSizeMake(200, 800);
scrollView.backgroundColor = [UIColor greenColor];
[self.view addSubview:scrollView];
[RACObserve(scrollView, contentOffset) subscribeNext:^(id x) {
    NSLog(@"success");
}];
```

如果你好好回顾下KVO的写法会不会更加感叹RAC的强大呢？

总结

RAC的东西很多，但相信这篇文章给大家入门还是可以的，这里介绍了RAC得基本使用方法，但是还有很多好用的东西比如信号都没介绍，甚至除了subscribeNext还有subscribeComplete和subscribeError，这些都是我给自己留得坑，以后也会发文介绍给大家。也欢迎来我的博客 (<http://cbsfly.github.io/>)看看。

随笔 (/nb/2889751)

举报文章 © 著作权归作者所有



cbsfly_iDev (/u/4daf2d76e124)

写了 13984 字，被 319 人关注，获得了 285 个喜欢
(/u/4daf2d76e124)

+ 关注

一个正在学习的iOS开发者 个人主页: <http://cbsfly.github.io/>

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持

喜欢 (/sign_in) | 136



更多分享

(<http://cw.b.assets.jianshu.io/notes/images/2713193/>)

[登录 \(/sign\)](#) 后发表评论

28条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



linc_ (/u/1426877f5f3d)

20楼 · 2017.02.17 17:07

[\(/u/1426877f5f3d\)](#)
写的不错

1人赞 回复



tausha (/u/5ccca04308e5)

2楼 · 2016.05.12 23:20

[\(/u/5ccca04308e5\)](#)
支持一下

赞 回复



4bd903b8c237 (/u/4bd903b8c237)

3楼 · 2016.05.17 09:50

[\(/u/4bd903b8c237\)](#)
如果是自定义的键盘 这个方法不起作用怎么解决？？
[[self.textField rac_textSignal] subscribeNext:^(id x) {
NSLog(@"%@",x);
}];

赞 回复

VincentPeng (/u/96f06ec4e80c): @4bd903b8c237 (/users/4bd903b8c237) 这个API是针对系统的UI控件定制的Cocoa框架，如果是自定义的键盘，就需要自己搭建信号和接收者

2016.07.07 15:47 回复

添加新评论



MyLikeiOS (/u/8c3fa9cdf022)

4楼 · 2016.05.27 12:45

[\(/u/8c3fa9cdf022\)](#)
深度不够，不太深刻，记不下来。不过支持下。

赞 回复

大了个涛 (/u/a2401c61d8f8): @MyLikeiOS (/users/8c3fa9cdf022) 文章题目就是说适合新手，作为入门学习文章已经很好了

2016.10.17 17:20 回复

Nick_Nickname (/u/b366a07517eb): @MyLikeiOS (/users/8c3fa9cdf022) 文章题目就是说适合新手,作为入门学习文章已经很好了+1

2016.11.04 11:33 回复

凯凯kikis (/u/2d19108128c1): 文章题目就是说适合新手,作为入门学习文章已经很好了+2

2016.12.08 15:57 回复

添加新评论 | 还有1条评论，展开查看



Joneding (/u/f706d9e98c7b)

5楼 · 2016.06.03 10:02

(/u/f706d9e98c7b)

给Label加入轻拍手势,使用RAC 点击Label没有反应 这是什么原因

```
UITapGestureRecognizer *tap = [[UITapGestureRecognizer alloc] init];
```

```
[self.textLabel addGestureRecognizer:tap];
```

```
[[tap rac_gestureSignal] subscribeNext:^(id x) {
```

```
NSLog(@"tap");
```

```
}};
```

👍 赞 💬 回复

3b442fc8a930 (/u/3b442fc8a930): @Joneding (/users/f706d9e98c7b)

userInteractionEnabled = YES

2016.07.16 03:30 💬 回复

✍️ 添加新评论



LS_Developer (/u/b39b9cf73968)

6楼 · 2016.07.28 17:53

(/u/b39b9cf73968)

大神, 请问rac_signalToSelector这个方法如何监听有返回值的代理方法呢? 比如说tableView的代理方法

👍 赞 💬 回复



断雪残梦 (/u/f003d5cbd5b3)

7楼 · 2016.08.03 16:44

(/u/f003d5cbd5b3)



👍 赞 💬 回复



翀鹰女孩 (/u/a3b4b4f29be0)

8楼 · 2016.09.20 18:01

(/u/a3b4b4f29be0)

你好 请问有demo吗, 能发我一下吗? 2807338860@qq.com 非常感谢了

👍 赞 💬 回复



选一个昵称也被使用了 (/u/e06d3e20bfe5)

9楼 · 2016.09.26 18:10

(/u/e06d3e20bfe5)

挺好的, 适合初学者入门了解RAC

👍 赞 💬 回复



FO简叔 (/u/2b7ee5592789)

10楼 · 2016.10.10 22:42

(/u/2b7ee5592789)

想问下这个创建的通知是否需要移除...


👍 赞 💬 回复

无夜之星辰 (/u/4212f351f6b5): @FO简叔 (/users/2b7ee5592789) 文中说的不需要

2017.02.10 20:16 💬 回复

FO简叔 (/u/2b7ee5592789): @无夜之星辰 (/users/4212f351f6b5) 我遇到了这种情况, 注册一个通知, 在其他多个界面监听, 就算移除了通知都会调用那些界面的方法, 移除不掉一样。

2017.02.10 20:22 💬 回复

 添加新评论



大了个涛 (/u/a2401c61d8f8)

11楼 · 2016.10.17 17:20

(/u/a2401c61d8f8)

特别适合初学者,比看那些照搬概念文章的清晰多了,感谢你的分享 😊

 赞  回复



dongwenbo (/u/e53d385ec9f8)

12楼 · 2016.11.15 09:00

(/u/e53d385ec9f8)

rac这么强大!

 赞  回复



梁同桌 (/u/b9e84b0af9c5)


13楼 · 2016.11.17 22:54

(/u/b9e84b0af9c5)

RAC 我猜底层用 KVO监听属性变化,然后用Block回调。

 赞  回复

海子云 (/u/34f8a1d69c11): 不是, 代码都在本地自己看一下

2017.02.04 19:42  回复

 添加新评论



随意_M (/u/5074ced291c7)

14楼 · 2016.11.25 14:59

(/u/5074ced291c7)

请问在哪儿看的直播,叶大是??? 我也想看看,最近项目要用RAC+MVVM 🤪 疯狂学习中

 赞  回复



孤胆走天涯 (/u/b3bcda7d8e9e)

15楼 · 2016.11.28 16:55

(/u/b3bcda7d8e9e)

挺好玩的

 赞  回复

1

2

下一页