

Cocos开发者平台

Cocos引擎中文官网

Cocos商店

a184251289

1 退出



全球最大苹果开发者中文社区

站内搜索

iOS 8 Swift App Store Apple Watch Metal Cocos引擎 手游

iOS开发
SwiftApp Store研究
游戏开发应用评测
苹果相关产品设计
营销推广Cocos引擎
业界动态WebApp
程序人生

论坛

代码

专题

活动

招聘

首页 > iOS开发

这样好用的ReactiveCocoa, 根本停不下来

2015-08-20 11:38 编辑: suiling 分类: iOS开发 来源: 空之境界投稿

12

6197

ReactiveCocoa 编程框架

招聘信息: C++程序员

本文是投稿文章, 作者: 空之境界 ([博客](#))

前戏

我个人非常推崇ReactiveCocoa, 它就像中国的太极, 太极生两仪, 两仪生四象, 四象生八卦, 八卦生万物。

ReactiveCocoa是一个高度抽象的编程框架, 它真的很抽象, 初看你不知道它是要干嘛的, 等你用上了之后, 就发现, 有了它你是想干嘛就干嘛, 编码从未如此流畅。

在此我不会讲ReactiveCocoa的原理, 因为不能讲明白的才叫抽象。我也不会提及相关概念。我只是让你看看我用着它是有多爽。

代码的四十八手

察值

你别动, 你一动我就知道。

```
1 @weakify(self);
2 [RACObserve(self, value) subscribeNext:^(NSString* x) {
3     @strongify(self);
4     NSLog(@"你动了");
5 }];
```

单边

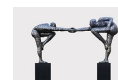
热门资讯

**8行代码教你搞定导航控制器全屏滑动返回效**

点击量 10102

**不说鸡汤, 谈谈现实: 半路学编程能成大牛**

点击量 7896

**UIWebView与JS的深度交互**

点击量 7803

**超详细! iOS 并发编程之 Operation Queues**

点击量 6732

**打造安全的App! iOS 安全系列之 HTTPS**

点击量 6678

**为什么程序员的业余项目大多都死了?**

点击量 6195

**为什么技术总是被轻视? 不服来辩!**

点击量 6127

**一文让你彻底了解iOS 字体相关知识**

点击量 5688

**深入理解Objective-C: 方法缓存**

点击量 4554

**我的iOS高效编程秘诀 - 坚持编程习惯**

点击量 4246

综合评论

我也要给我

047691641 评论了 七夕我们在一起, 送你12只萌萌哒的COCO熊! 速抢...

Objective-C runtime真心很强大

yiwaji 评论了 深入理解Objective-C: 方法缓存

什么都没有啊

xxfftywc90 评论了 三年一个人使用虚幻引擎(UDK)开发的一个游戏心路

这个看起来很好用 不过现在 用的xcode6.3 需要升级才能用

hall_of_fame 评论了 iOS 9学习系列:

你唱歌，我就跳舞。

textField的内容长度隐射成BOOL值，绑定到confirmButton的enable属性上面，当textField输入内容不为空的时候，confirmButton的enable = YES。

```
1 RACSignal *signalA = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2 [subscriber sendNext:@"唱歌"];
3 [subscriber sendCompleted];
4 return nil;
5 ]];
6 RAC(self, value) = [signalA map:^(NSString* value) {
7     if ([value isEqualToString:@"唱歌"]) {
8         return @"跳舞";
9     }
10    return @"";
11 }];
```

双边

你向西，他就向东，他向左，你就向右。

```
1 RACChannelTerminal *channelA = RACChannelTo(self, valueA);
2 RACChannelTerminal *channelB = RACChannelTo(self, valueB);
3 [[channelA map:^(NSString *value) {
4     if ([value isEqualToString:@"西"]) {
5         return @"东";
6     }
7     return value;
8 }] subscribe:channelB];
9 [[channelB map:^(NSString *value) {
10    if ([value isEqualToString:@"左"]) {
11        return @"右";
12    }
13    return value;
14 }] subscribe:channelA];
15 [[RACObserve(self, valueA) filter:^(BOOL(id value) {
16    return value ? YES : NO;
17 }] subscribeNext:^(NSString* x) {
18    NSLog(@"你向%@", x);
19 }]];
20 [[RACObserve(self, valueB) filter:^(BOOL(id value) {
21    return value ? YES : NO;
22 }] subscribeNext:^(NSString* x) {
23    NSLog(@"他向%@", x);
24 }]];
25 self.valueA = @"西";
26 self.valueB = @"左";
```

```
1 2015-08-15 20:14:46.544 Test[2440:99901] 你向西
2 2015-08-15 20:14:46.544 Test[2440:99901] 他向东
3 2015-08-15 20:14:46.545 Test[2440:99901] 他向左
4 2015-08-15 20:14:46.545 Test[2440:99901] 你向右
```

代理

你是程序员，你帮我写个app吧。

```
1 @protocol Programmer
2 - (void)makeAnApp;
3 @end
4 RACSignal *ProgrammerSignal =
5 [self rac_signalForSelector:@selector(makeAnApp)
6     fromProtocol:@protocol(Programmer)];
7 [ProgrammerSignal subscribeNext:^(RACTuple* x) {
8     NSLog(@"花了一个月，app写好了");
9 }];
10 [self makeAnApp];

1 2015-08-15 20:46:45.720 Test[2817:114564] 花了一个月，app写好了
```

广播

知道你的频道，我就能听到你了。

```
1 [[NSNotificationCenter defaultCenter] rac_addObserverForName:@"代码之道频道"
```

UIStackView如何让你...

遇到这种情况，不要着急赞成，也不要着急否认。即使你是一个有过几个成功
jarod姜赫 评论了 大神支招：如何应对老板干涉游戏设计问题

牛，都说到点上了。

jarod姜赫 评论了 国外游戏开发者：创业路上9大禁区不可进

楼主一生平安!!!!

向宏simida 评论了 源码推荐(8.18): 类似支付宝余额数字动画——贝...

6666666

fu779115154 评论了 阿峥教你实现

UITableView循环利用

作者的编程习惯是值得学习的，而且这样做也能提高编码效率。这样的习惯也
icebergcwp1990 评论了 我的iOS高效编程秘诀——坚持编程习惯

。。。。意义不明

leonajkl 评论了 阿峥教你实现

UITableView循环利用

相关帖子

关于AFNnetWork的一个编译报错

中国省市plist文件的解析

PUParticleSystem3D 显示出方块 求解决

求助，tableView accessoryView怎么实现的

求使用3.7.1新骨骼动画示例 感恩

PUParticleSystem3D 显示出方块 求解决

关于服务器返回json格式问题请教

Ble4.0问题

开发者账号续费后，iTunes没有把钱支付到公司银行账户上去

```
2 NSLog(@"技巧: %@", x.userInfo[@"技巧"]);
3 };
4 [[NSNotificationCenter defaultCenter] postNotificationName:@"代码之道频道" obj
```

```
1 | 2015-08-15 20:41:15.786 Test[2734:111505] 技巧: 用心写
```

连接

生活是一个故事接一个故事。

```
1 RACSignal *signalA = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2 [subscriber sendNext:@"我恋爱啦"];
3 [subscriber sendCompleted];
4 return nil;
5 ];
6 RACSignal *signalB = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
7 [subscriber sendNext:@"我结婚啦"];
8 [subscriber sendCompleted];
9 return nil;
10 ];
11 [[signalA concat:signalB] subscribeNext:^(id x) {
12 NSLog(@"%@", x);
13 }];
```

```
1 | 2015-08-15 12:19:46.707 Test[1845:64122] 我恋爱啦
2 | 2015-08-15 12:19:46.707 Test[1845:64122] 我结婚啦
```

合并

污水都应该流入污水处理厂被处理。

```
1 RACSignal *signalA = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2 [subscriber sendNext:@"纸厂污水"];
3 return nil;
4 ];
5 RACSignal *signalB = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
6 [subscriber sendNext:@"电镀厂污水"];
7 return nil;
8 ];
9 [[RACSignal merge:[signalA, signalB]] subscribeNext:^(id x) {
10 NSLog(@"处理%@", x);
11 }];
```

```
1 | 2015-08-15 12:10:05.371 Test[1770:60147] 处理纸厂污水
2 | 2015-08-15 12:10:05.372 Test[1770:60147] 处理电镀厂污水
```

组合

你是红的, 我是黄的, 我们就是红黄的, 你是白的, 我没变, 我们是白黄的。

```
1 RACSignal *signalA = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2 [subscriber sendNext:@"红"];
3 [subscriber sendNext:@"白"];
4 return nil;
5 ];
6 RACSignal *signalB = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
7 [subscriber sendNext:@"白"];
8 return nil;
9 ];
10 [[RACSignal combineLatest:[signalA, signalB]] subscribeNext:^(RACTuple* x)
11 RACTupleUnpack(NSSString *stringA, NSString *stringB) = x;
12 NSLog(@"我们是%@%@"的", stringA, stringB);
13 }];
```

```
1 | 2015-08-15 12:14:19.837 Test[1808:62042] 我们就是红黄的
2 | 2015-08-15 12:14:19.837 Test[1808:62042] 我们是白黄的
```

压缩

你是红的, 我是黄的, 我们就是红黄的, 你是白的, 我没变, 哦, 那就等我变了再说吧。

```
1 RACSignal *signalA = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2 [subscriber sendNext:@"红"];
3 [subscriber sendNext:@"白"];
```

微博



CocoaChina

已关注

【iOS 9学习系列: 如何使用ATS提高应用的安全性】在过去几年中, 对web安全的注意和它的重要性都在充分的稳定增长。在今天的WWDC大会上, Apple明确表示了它打算以身作则的通过一个新特性来提高操作系统的安全性, 它就是App Transport Security. 详见: <http://t.cn/RLDe838>



19分钟前

转发(3) | 评论

【听故事搞懂多线程开发-- 屎壳郎老板和它的收费公厕】刚接触开发的童鞋, 在学习多线程的时候, 往往会比较吃力. 因为这家伙比较抽象, 有的时候



```
4         return nil;
5     }];
6     RACSignal *signalB = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
7         [subscriber sendNext:@"白"];
8         return nil;
9     ]];
10    [[signalA zipWith:signalB] subscribeNext:^(RACTuple* x) {
11        RACTupleUnpack(NSString *stringA, NSString *stringB) = x;
12        NSLog(@"我们是%@%@" , stringA, stringB);
13    }];
```

```
1 | 2015-08-15 20:34:24.274 Test[2660:108483] 我们是红白的
```

映射

我可以点石成金。

```
1     RACSignal *signal = [[RACSignal createSignal:^(RACDisposable * (idsubscriber)
2         [subscriber sendNext:@"石"];
3         return nil;
4     ] map:^(NSString* value) {
5         if ([value isEqualToString:@"石"]) {
6             return @"金";
7         }
8         return value;
9     }];
10    [signal subscribeNext:^(id x) {
11        NSLog(@"%@", x);
12    }];
```

```
1 | 2015-08-16 20:00:12.853 Test[740:15871] 金
```

归约

糖加水变成糖水。

```
1     RACSignal *sugarSignal = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
2         [subscriber sendNext:@"糖"];
3         return nil;
4     ]];
5     RACSignal *waterSignal = [RACSignal createSignal:^(RACDisposable * (idsubscriber)
6         [subscriber sendNext:@"水"];
7         return nil;
8     ]];
9     [[RACSignal combineLatest:@[sugarSignal, waterSignal] reduce:^(id (NSString*)
10         return [sugar stringByAppendingString:water];
11     ] subscribeNext:^(id x) {
12         NSLog(@"%@", x);
13     }];
```

```
1 | 2015-08-16 20:07:00.356 Test[807:19177] 糖水
```

过滤

未满十八岁, 禁止进入。

```
1     [[RACSignal createSignal:^(RACDisposable * (idsubscriber) {
2         [subscriber sendNext:@(15)];
3         [subscriber sendNext:@(17)];
4         [subscriber sendNext:@(21)];
5         [subscriber sendNext:@(14)];
6         [subscriber sendNext:@(30)];
7         return nil;
8     } filter:^(NSNumber* value) {
9         return value.integerValue >= 18;
10    } subscribeNext:^(id x) {
11        NSLog(@"%@", x);
12    }];
```

```
1 | 2015-08-16 20:11:20.071 Test[860:21214] 21
2 | 2015-08-16 20:11:20.071 Test[860:21214] 30
```

扁平

打蛋液, 煎鸡蛋, 上盘。

```

1  [[[RACSignal createSignal:^RACDisposable *(idsubscriber) {
2      NSLog(@"打蛋液");
3      [subscriber sendNext:@"蛋液"];
4      [subscriber sendCompleted];
5      return nil;
6  }] flattenMap:^RACStream *(NSString* value) {
7      return [RACSignal createSignal:^RACDisposable *(idsubscriber) {
8          NSLog(@"把%@倒进锅里面煎", value);
9          [subscriber sendNext:@"煎蛋"];
10         [subscriber sendCompleted];
11         return nil;
12     }]];
13  }] flattenMap:^RACStream *(NSString* value) {
14      return [RACSignal createSignal:^RACDisposable *(idsubscriber) {
15          NSLog(@"把%@装到盘里", value);
16          [subscriber sendNext:@"上菜"];
17          [subscriber sendCompleted];
18          return nil;
19      }]];
20  }] subscribeNext:^(id x) {
21      NSLog(@"%@", x);
22  }];

```

```

1  2015-08-16 20:39:34.786 Test[1226:34386] 打蛋液
2  2015-08-16 20:39:34.787 Test[1226:34386] 把蛋液倒进锅里面煎
3  2015-08-16 20:39:34.787 Test[1226:34386] 把煎蛋装到盘里
4  2015-08-16 20:39:34.787 Test[1226:34386] 上菜

```

秩序

把大象塞进冰箱只需要三步：打开冰箱门，把大象塞进冰箱，关上冰箱门。

```

1  [[[RACSignal createSignal:^RACDisposable *(idsubscriber) {
2      NSLog(@"打开冰箱门");
3      [subscriber sendCompleted];
4      return nil;
5  }] then:^RACSignal *{
6      return [RACSignal createSignal:^RACDisposable *(idsubscriber) {
7          NSLog(@"把大象塞进冰箱");
8          [subscriber sendCompleted];
9          return nil;
10     }]];
11  }] then:^RACSignal *{
12      return [RACSignal createSignal:^RACDisposable *(idsubscriber) {
13          NSLog(@"关上冰箱门");
14          [subscriber sendCompleted];
15          return nil;
16     }]];
17  }] subscribeCompleted:^(
18      NSLog(@"把大象塞进冰箱了");
19  );

```

```

1  2015-08-16 20:45:27.724 Test[1334:37870] 打开冰箱门
2  2015-08-16 20:45:27.725 Test[1334:37870] 把大象塞进冰箱
3  2015-08-16 20:45:27.725 Test[1334:37870] 关上冰箱门
4  2015-08-16 20:45:27.726 Test[1334:37870] 把大象塞进冰箱了

```

命令

我命令你马上投降。

```

1  RACCommand *aCommand = [[RACCommand alloc] initWithSignalBlock:^RACSignal *(
2      return [RACSignal createSignal:^RACDisposable *(idsubscriber) {
3          NSLog(@"我投降了");
4          [subscriber sendCompleted];
5          return nil;
6      }]];
7  ];
8  [aCommand execute:nil];

```

```

1  2015-08-16 20:54:32.492 Test[1450:41849] 我投降了

```

延迟

等等我，我还有10秒钟就到了。

```

1  [[[RACSignal createSignal:^RACDisposable *(idsubscriber) {
2      NSLog(@"等等我，我还有10秒钟就到了");

```

```
3         [subscriber sendNext:nil];
4         [subscriber sendCompleted];
5         return nil;
6     } delay:10] subscribeNext:^(id x) {
7         NSLog(@"我到了");
8     }];
```

```
1 2015-08-16 21:00:57.622 Test[1619:45924] 等等我, 我还有10秒钟就到了
2 2015-08-16 21:01:07.624 Test[1619:45924] 我到了
```

重放

一次制作, 多次观看。

```
1 RACSignal *replaySignal = [[RACSignal createSignal:^(RACDisposable *(idsubsc
2     NSLog(@"大导演拍了一部电影《我的男票是程序员》");
3     [subscriber sendNext:@"《我的男票是程序员》"];
4     return nil;
5 ] replay];
6 [replaySignal subscribeNext:^(id x) {
7     NSLog(@"小明看了%@", x);
8 }];
9 [replaySignal subscribeNext:^(id x) {
10    NSLog(@"小红也看了%@", x);
11 }];
```

```
1 2015-08-16 21:18:38.002 Test[1854:54712] 大导演拍了一部电影《我的男票是程序员》
2 2015-08-16 21:18:38.004 Test[1854:54712] 小明看了《我的男票是程序员》
3 2015-08-16 21:18:38.004 Test[1854:54712] 小红也看了《我的男票是程序员》
```

定时

每隔8个小时服一次药。

```
1 [[RACSignal interval:60*60*8 onScheduler:[RACScheduler mainThreadScheduler]]
2     NSLog(@"吃药");
3 ]];
```

超时

等了你一个小时了, 你还没来, 我走了。

```
1 [[[[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
2     [[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
3         NSLog(@"我快到了");
4         [subscriber sendNext:nil];
5         [subscriber sendCompleted];
6         return nil;
7     }] delay:60*70] subscribeNext:^(id x) {
8         [subscriber sendNext:nil];
9         [subscriber sendCompleted];
10    }];
11    return nil;
12 }] timeout:60*60 onScheduler:[RACScheduler mainThreadScheduler]] subscribeE
13     NSLog(@"等了你一个小时了, 你还没来, 我走了");
14 ]];
```

```
1 2015-08-16 21:40:09.068 Test[2041:64720] 我快到了
2 2015-08-16 22:40:10.048 Test[2041:64720] 等了你一个小时了, 你还没来, 我走了
```

重试

成功之前可能需要数百次失败。

```
1 block int failedCount = 0;
2 [[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
3     if (failedCount < 100) {
4         failedCount++;
5         NSLog(@"我失败了");
6         [subscriber sendError:nil];
7     } else {
8         NSLog(@"经历了数百次失败后");
9         [subscriber sendNext:nil];
10    }
11    return nil;
12 }];
```



```

12     }} retry] subscribeNext:^(id x) {
13         NSLog(@"终于成功了");
14     }};

1 2015-08-16 21:59:07.159 Test[2411:77080] 我失败了
2 2015-08-16 21:59:07.159 Test[2411:77080] 我失败了
3 2015-08-16 21:59:07.159 Test[2411:77080] 我失败了
4 2015-08-16 21:59:07.159 Test[2411:77080] 我失败了
5 2015-08-16 21:59:07.160 Test[2411:77080] 我失败了
6 2015-08-16 21:59:07.160 Test[2411:77080] 我失败了
7 2015-08-16 21:59:07.161 Test[2411:77080] 我失败了
8 2015-08-16 21:59:07.162 Test[2411:77080] 我失败了
9 ...
10 2015-08-16 21:59:07.162 Test[2411:77080] 我失败了
11 2015-08-16 21:59:07.163 Test[2411:77080] 我失败了
12 2015-08-16 21:59:07.163 Test[2411:77080] 我失败了
13 2015-08-16 21:59:07.163 Test[2411:77080] 我失败了
14 2015-08-16 21:59:07.164 Test[2411:77080] 我失败了
15 2015-08-16 21:59:07.164 Test[2411:77080] 我失败了
16 2015-08-16 21:59:07.164 Test[2411:77080] 我失败了
17 2015-08-16 21:59:07.165 Test[2411:77080] 经历了数百次失败后
18 2015-08-16 21:59:07.165 Test[2411:77080] 终于成功了

```

节流

不好意思, 这里一秒钟只能通过一个人。

```

1 [[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
2     [subscriber sendNext:@"旅客A"];
3     dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(1 * NSEC_PER_
4         [subscriber sendNext:@"旅客B"];
5     });
6     dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(2 * NSEC_PER_
7         [subscriber sendNext:@"旅客C"];
8         [subscriber sendNext:@"旅客D"];
9         [subscriber sendNext:@"旅客E"];
10    });
11    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(3 * NSEC_PER_
12        [subscriber sendNext:@"旅客F"];
13    });
14    return nil;
15 }} throttle:1] subscribeNext:^(id x) {
16     NSLog(@"%@通过了", x);
17 });

```

```

1 2015-08-16 22:08:45.677 Test[2618:83764] 旅客A
2 2015-08-16 22:08:46.737 Test[2618:83764] 旅客B
3 2015-08-16 22:08:47.822 Test[2618:83764] 旅客E
4 2015-08-16 22:08:48.920 Test[2618:83764] 旅客F

```

条件

直到世界的尽头才能把我们分开。

```

1 [[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
2     [[RACSignal interval:1 onScheduler:[RACSchedulder mainThreadScheduler]]
3         [subscriber sendNext:@"直到世界的尽头才能把我们分开"];
4     });
5     return nil;
6 }} takeUntil:[RACSignal createSignal:^(RACDisposable *(idsubscriber) {
7     dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(5 * NSEC_PER_
8         NSLog(@"世界的尽头到了");
9         [subscriber sendNext:@"世界的尽头到了"];
10    });
11    return nil;
12 }}] subscribeNext:^(id x) {
13     NSLog(@"%@", x);
14 });

```

```

1 2015-08-16 22:17:22.648 Test[2766:88737] 直到世界的尽头才能把我们分开
2 2015-08-16 22:17:23.648 Test[2766:88737] 直到世界的尽头才能把我们分开
3 2015-08-16 22:17:24.645 Test[2766:88737] 直到世界的尽头才能把我们分开
4 2015-08-16 22:17:25.648 Test[2766:88737] 直到世界的尽头才能把我们分开
5 2015-08-16 22:17:26.644 Test[2766:88737] 直到世界的尽头才能把我们分开
6 2015-08-16 22:17:26.645 Test[2766:88737] 世界的尽头到了

```

完事

ReactiveCocoa是如此优雅, 一旦使用, 根本停不下来, 上面也只是它的一角冰山, 但愿我能挑起你的兴趣。



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号：CocoaChina

我要投稿

收藏文章

分享到：

52

上一篇：源码推荐(8.20)：仿evernote和微信标签效果，一个向上划动手势转场动画

下一篇：深入理解Objective-C：方法缓存

相关资讯

RACSignal的Subscription深入分析

【长篇高能】ReactiveCocoa 和 MVVM 入门

ReactiveCocoa入门教程：第一部分

ReactiveCocoa2实战

说说ReactiveCocoa 2

ReactiveCocoa自述：工作原理和应用

CoreData 与 ReactiveCocoa 混用时要注意的线程问题

ReactiveCocoa 用 RACSignal 替代 Delegate

基于AFNetworking2.0和ReactiveCocoa2.1的iOS REST



坚持1个月 看美剧不用字幕

每天45分钟 30天见证你的英语奇迹

立即行动 >

文章评论 (2)



canyu9512 2015-08-21 09:43:08

打算研究下，好像还挺不错的

支持(0)

回复(0)



庸者的救赎 2015-08-21 08:25:31

一直使用，从未有哪个项目里面不用的，强烈推荐

支持(0)

回复(0)

对这篇文章有什么感想，写一下吧.....

发表评论

[关于我们](#) [广告合作](#) [联系我们](#) [Cocos商店](#)

©2015 Chukong Technologies, Inc.

京ICP备 11006519号

京ICP证 100954号

京公网安备11010502020289



京网文[2012]0426-138号