深入浅出RxJava（一：基础篇）

标签： RxAndroid RxJava Android ReactiveFunctionProg 响应式函数编程

2014-12-09 23:47 65454人阅读 评论(56) 收藏 举报

分类： Android（14） RxJava（6）

目录(?) [+]

原文链接

RxJava正在Android开发者中变的越来越流行。唯一的问题就是上手不容易，尤其是大部分人之前都是使用命令式编程语言。但是一旦你弄明白了，你就会发现RxJava真是太棒了。
这里仅仅是帮助你了解RxJava，整个系列共有四篇文章，希望你看完这四篇文章之后能够了解RxJava背后的思想，并且喜欢上RxJava。

基础

RxJava最核心的两个东西是Observables（被观察者，事件源）和Subscribers（观察者）。Observables发出一系列事件，Subscribers处理这些事件。这里的事件可以是任何你感兴趣的东西（触摸事件，web接口调用返回的数据。。。）

一个Observable可以发出零个或者多个事件，知道结束或者出错。每发出一个事件，就会调用它的Subscriber的onNext方法，最后调用Subscriber.onNext()或者Subscriber.onError()结束。

Rxjava的看起来很想设计模式中的观察者模式，但是有一点明显不同，那就是如果一个Observerble没有任何的的Subscriber，那么这个Observable是不会发出任何事件的。

Hello World

创建一个Observable对象很简单，直接调用Observable.create即可

```
[java]
Observable<String> myObservable = Observable.create(
    new Observable.OnSubscribe<String>() {
        @Override
        public void call(Subscriber<? super String> sub) {
            sub.onNext("Hello, world!");
            sub.onCompleted();
        }
    }
);
```

这里定义的Observable对象仅仅发出一个Hello World字符串，然后就结束了。接着我们创建一个Subscriber来处理Observable对象发出的字符串。

```
[java]
Subscriber<String> mySubscriber = new Subscriber<String>() {
    @Override
    public void onNext(String s) { System.out.println(s); }

    @Override
    public void onCompleted() { }

    @Override
    public void onError(Throwable e) { }
```

hi大头鬼hi

访问： 285478次
积分： 1589
等级：
排名： 第15832名

原创： 14篇 转载： 0篇
译文： 12篇 评论： 174条


```
10. };
```

这里subscriber仅仅就是打印observable发出的字符串。通过subscribe函数就可以将我们定义的myObservable对象和mySubscriber对象关联起来，这样就完成了subscriber对observable的订阅。

```
[java]
01. myObservable.subscribe(mySubscriber);
```

一旦mySubscriber订阅了myObservable，myObservable就是调用mySubscriber对象的onNext和onComplete方法，mySubscriber就会打印出Hello World！

更简洁的代码

是不是觉得仅仅为了打印一个hello world要写这么多代码太啰嗦？我这里主要是为了展示RxJava背后的原理而采用了这种比较啰嗦的写法，RxJava其实提供了很多便捷的函数来帮助我们减少代码。

首先来看看如何简化Observable对象的创建过程。RxJava内置了很多简化创建Observable对象的函数，比如Observable.just就是用来创建只发出一个事件就结束的Observable对象，上面创建Observable对象的代码可以简化为一行

```
[java]
01. Observable<String> myObservable = Observable.just("Hello, world!");
```

接下来看看如何简化Subscriber，上面的例子中，我们其实并不关心OnComplete和OnError，我们只需要在onNext的时候做一些处理，这时候就可以使用Action1类。

```
[java]
01. Action1<String> onNextAction = new Action1<String>() {
02.     @Override
03.     public void call(String s) {
04.         System.out.println(s);
05.     }
06. };
```

subscribe方法有一个重载版本，接受三个Action1类型的参数，分别对应OnNext，OnComplete， OnError函数。

```
[java]
01. myObservable.subscribe(onNextAction, onErrorAction, onCompleteAction);
```

这里我们并不关心onError和onComplete，所以只需要第一个参数就可以

```
[java]
01. myObservable.subscribe(onNextAction);
02. // Outputs "Hello, world!"
```

上面的代码最终可以写成这样

```
[java]
01. Observable.just("Hello, world!")
02.     .subscribe(new Action1<String>() {
03.         @Override
04.         public void call(String s) {
05.               System.out.println(s);
06.         }
07.     });
```

使用java8的lambda可以使代码更简洁

```
[java]
01. Observable.just("Hello, world!")
02.     .subscribe(s -> System.out.println(s));
```

Android开发中，强烈推荐使用retrolambda这个gradle插件，这样你就可以在你的代码中使用lambda了。

变换

让我们做一些更有趣的事情吧！

比如我想在hello world中加上我的签名，你可能会想到去修改Observable对象：

[java]

```
Observable.just("Hello, world! -Dan")
    .subscribe(s -> System.out.println(s));
```

如果你能够改变Observable对象，这当然是可以的，但是如果你不能修改Observable对象呢？比如Observable对象是第三方库提供的？比如我的Observable对象被多个Subscriber订阅，但是我只想在对某个订阅者做修改呢？那么在Subscriber中对事件进行修改怎么样呢？比如下面的代码：

[java]

```
Observable.just("Hello, world!")
    .subscribe(s -> System.out.println(s + " -Dan"));
```

这种方式仍然不能让人满意，因为我希望我的Subscribers越轻量越好，因为我有可能会在mainThread中运行subscriber。另外，根据响应式函数编程的概念，Subscribers更应该做的事情是“响应”，响应Observable发出的事件，而不是去修改。如果我能在某些中间步骤中对“Hello World！”进行变换是不是很酷？

操作符（**Operators**）

操作符就是为了解决对Observable对象的变换的问题，操作符用于在Observable和最终的Subscriber之间修改Observable发出的事件。RxJava提供了很多很有用的操作符。

比如map操作符，就是用来把把一个事件转换为另一个事件的。

[java]

```
Observable.just("Hello, world!")
  .map(new Func1<String, String>() {
      @Override
      public String call(String s) {
          return s + " -Dan";
      }
  })
  .subscribe(s -> System.out.println(s));
```

使用lambda可以简化为

[java]

```
Observable.just("Hello, world!")
    .map(s -> s + " -Dan")
    .subscribe(s -> System.out.println(s));
```

是不是很酷？map()操作符就是用于变换Observable对象的，map操作符返回一个Observable对象，这样就可以实现链式调用，在一个Observable对象上多次使用map操作符，最终将最简洁的数据传递给Subscriber对象。

map操作符进阶

map操作符更有趣的一点是它不必返回Observable对象返回的类型，你可以使用map操作符返回一个发出新的数据类型的observable对象。

比如上面的例子中，subscriber并不关心返回的字符串，而是想要字符串的hash值

[java]

```
Observable.just("Hello, world!")
    .map(new Func1<String, Integer>() {
        @Override
        public Integer call(String s) {
            return s.hashCode();
        }
    })
    .subscribe(i -> System.out.println(Integer.toString(i)));
```

很有趣吧？我们初始的Observable返回的是字符串，最终的Subscriber收到的却是Integer，当然使用lambda可以进一步简化代码：

[java]

```
Observable.just("Hello, world!")
    .map(s -> s.hashCode())
    .subscribe(i -> System.out.println(Integer.toString(i)));
```

前面说过，Subscriber做的事情越少越好，我们再增加一个map操作符

[java]

```
Observable.just("Hello, world!")
    .map(s -> s.hashCode())
    .map(i -> Integer.toString(i))
    .subscribe(s -> System.out.println(s));
```

不服？

是不是觉得我们的例子太简单，不足以说服你？你需要明白下面的两点:

1.Observable和Subscriber可以做任何事情

Observable可以是一个数据库查询，Subscriber用来显示查询结果；Observable可以是屏幕上的点击事件，Subscriber用来响应点击事件；Observable可以是一个网络请求，Subscriber用来显示请求结果。

2.Observable和Subscriber是独立于中间的变换过程的。

在Observable和Subscriber中间可以增减任何数量的map。整个系统是高度可组合的，操作数据是一个很简单的过程。

上一篇 android log你不知道的小技巧
下一篇 Otto使用入门

我的同类文章

Android（14） RxJava（6）

- DynamicAPK基本概念 2015-12-06 阅读 1350
- 如何实现携程动态加载插件... 2015-11-10 阅读 1642
- android-gradle-深入浅出四:... 2015-01-09 阅读 1992
- 深入浅出Android Gradle构... 2014-12-27 阅读 3577
- Otto使用入门 2014-12-19 阅读 4144
- Android热更新实现原理 2015-11-15 阅读 7787
- android-gradle-深入浅出-五... 2015-01-26 阅读 3058
- 深入浅出Android Gradle构... 2015-01-05 阅读 3773
- Otto源码分析 2014-12-19 阅读 1338

更多文章

主题推荐 android java android开发 编程语言 设计模式 对象 数据

猜你在找

Android设计模式精解(第7课) ：Adapter模式
Android设计模式精解(第10课) ：状态模式(State pat
Android必备的Java基础知识(三)
威哥全套Android开发课程【基础与UI技术】
Android基础视频教程
Android开发设计模式04
Android开发设计模式02
Android开发设计模式03
Android开发中的MVC设计模式
大话Android开发中的设计模式

查看评论

44楼 AndroidSummer 51分钟前发表

看了很多相关的文章

43楼 放风筝的骚年 昨天 09:2[illegible]发表

不明觉厉

42楼 walfud 4天前 18:56发表

感谢扔物线大神和 Bruce!
我也写了一系列源码分析, 欢迎交流:
'彻底搞懂 RxJava --- 基础篇' http://diordna.sinaapp.com/?p=896
后面还有: '彻底搞懂 RxJava --- 中级篇' 和 '彻底搞懂 RxJava --- 高级篇'

41楼 林深 2016-02-20 08:23发表

赞，学习了！

40楼 黄枫_ 2016-02-06 12:32发表

复杂的简单化，赞一个

39楼 孤心泣 2016-01-21 16:23发表

顶大头鬼同学！！！！！！！！！！！！

38楼 -琥珀川- 2016-01-16 11:24发表

mark

Re: 果冻虾仁 2016-02-04 22:24发表

回复u010032372：=_=

37楼 w7421 2016-01-13 16:25发表

mark~~

36楼 大-树 2016-01-13 09:44发表

mark 学习了

35楼 Joe_n 2016-01-12 12:44发表

建议lz 把lambda单独抽出来说，毕竟对于初学者来说，太抽象了，个人的建议，要想学rxjava 先把简单的evetbus弄清楚，再来学习rxjava 比较好。~_~

Re: _XuDaojie 2016-01-20 23:28发表

回复telencool：同意

34楼 Jasonez 2016-01-11 16:10发表

最后调用Subscriber.onNext()或者Subscriber.onError()结束。

应该将onNext改成onCompleted

33楼 Ners-曲 2016-01-08 23:21发表

登录下只为给个好评, 谢谢了

32楼 wenmin92 2016-01-04 16:39发表

看了几个讲 RxJava 比较有名的，这个是最简单、最明白的，赞！

31楼 xia 2015-12-28 12:37发表

@HI大头鬼,你好，方便加Q 2959945261 跟您请教一下关于Rxjava的一些问题吗，我这边代码中有一些疑惑，觉得您的博客写的非常好

30楼 xia 2015-12-28 12:36发表

@HI大头鬼,你好，方便加Q 2959945261 跟您请教一下关于Rxjava的一些问题吗，我这边代码中有一些疑惑，觉得您的博客写的非常好

29楼 kwxtx 2015-12-18 11:56发表

Rxjava的看起来很想设计模式中的观察者模式，但是有一点明显不同，那就是如果一个Observerble没有任何的的Subscriber，那么这个Observable是不会发出任何事件的。这句话怎么理解的？

Re: pzyoung 2015-12-27 14:47发表

回复kwxtx：看源码就知道了，所有的onNext逻辑都是在subscribe()那一刻触发的。

Re: 子墨_ 2015-12-26 17:08发表

回复kwxtx：个人理解啊，就像button没写click事件差不多

Re: wyk304443164 2015-12-22 09:45发表

回复kwxtx：我还没看Rx，我是这么理解的，如果没有观察者，那么被观察者不会发出，因为发出也没人接受

28楼 pscj 2015-12-10 16:08发表

"最后调用Subscriber.onNext()或者Subscriber.onError()结束。"
这一块应该是OnComplete()吧

27楼 我是东方谁是不败 2015-11-19 11:22发表

不错不错，看了这文章我才有动力去研究AndroidStudio怎么使用Lambda，终于找到AS使用Lambda最简单的方法了

26楼 Bran886 2015-11-03 15:13发表

6666666666666哈哈哈 受益匪浅

25楼 zoufuguo 2015-10-28 18:32发表

楼主，看过你的博文后受益匪浅，请问个问题，我如果想给Observable添加多个Subscriber要肿么添加呢，我直接写了两行
myObservable.subscribe (mysub1);
myObservable.subscribe(mysub2);
发现这样写mysub1和mysub2并没有各自独立开，还是先执行完mysub1再执行mysub2,请问我如果想让mysub1和mysub2各自独立线程运行需要怎么处理的，谢谢。

Re: 浮游生物_plankton 2015-12-08 16:24发表

回复zoufuguo：我测试了一下,注册了两个观察者,然后再onNext里面打印了System.currentTimeMillis().发现时间是一样的,应该证明是同时执行的吧,如果是一个先一个后的话,时间应该是错开的啊

24楼 zoufuguo 2015-10-28 18:27发表

楼主，我看上面只有一个subscribe（观察者）如果我想添加多个观察者的话是要怎么操作的呢，是写两个.subscribe(); 还是有其他办法的，我写了两个这个，发现两个观察者并没有相对线程独立，而是等一个执行完后，另一个才执行，求指教，谢谢。

23楼 zoufuguo 2015-10-28 18:26发表

楼主，我看上面只有一个subscribe（观察者）如果我想添加多个观察者的话是要怎么操作的呢，是写两个.subscribe(); 还是有其他办法的，我写了两个这个，发现两个观察者并没有相对线程独立，而是等一个执行完后，另一个才执行，求指教，谢谢。

22楼 liubaoyua 2015-10-16 09:45发表

有一个地方博主写错了。
博文的 "最后调用Subscriber.onNext()或者Subscriber.onError()结束。" 这里的 .onNext() 应该是 onComplete()

21楼 androidstackoverflow 2015-10-06 14:14发表

1.比如map操作符，就是用来把把一个事件转换为另一个事件的。应该将把把去掉一个。2.操作符就是为了解决对Observable对象的变换的问题。应该将其修改为……对象的变换问题。去掉第二个的。

20楼 androidstackoverflow 2015-10-06 13:59发表

Rxjava的看起来很想设计模式中的观察者模式，应该修改为Rxjava看起来很像设计模式中的观察者模式。将的去掉，将很想换成很像。

19楼 androidstackoverflow 2015-10-06 13:53发表

文章中的例子使用了Lambda表达式，对不知道的同学不好懂。联系换成java8之前的写法。

18楼 androidstackoverflow 2015-10-06 13:50发表

文章写得简单明了，思路清晰，很容易理解，尤其是将Observable翻译成:事件源，一下子，就好懂了很多。

17楼 androidstackoverflow 2015-10-06 13:49发表

应该在文章的开始说明一下 Subscriber 是 Observer 的实现类。这样子比较好。Action1和Action0有什么区别？是不是Action1可以泛型？

Re: 幻影浪子 2015-10-10 09:44发表

回复youyoumaomao：Subscriber 可不是 Observer 的实现类噢

16楼 androidstackoverflow 2015-10-06 13:47发表

每发出一个事件，就会调用它的Subscriber的onNext方法，最后调用Subscriber.onNext()或者Subscriber.onError()结束。 应该是最后调用Subscriber.onCompleted() 或者Subscriber.onError()结束。

15楼 androidstackoverflow 2015-10-06 13:45发表

知道结束或者出错，应该是直到。字打错了。

14楼 Cogitate_ 2015-09-16 11:48发表

一直不太明白，直到看了这篇文章

13楼 apple_zl 2015-08-20 00:35发表

不错，浅显易懂

12楼 wx7788250 2015-05-29 09:28发表

Observable.just("Hello, world! -Dan")
.subscribe(s -> System.out.println(s));

这里一定要使用lambda么?

Re: Justlove_DK 2015-08-19 11:35发表

回复wx7788250：不一定吧，这是java8的写法，6和7好像还没有lambda的吧

11楼 安卓 2015-04-28 09:45发表

每发出一个事件，就会调用它的Subscriber的onNext方法，最后调用Subscriber.onNext()或者Subscriber.onError()结束。

For each Subscriber it has, an Observable calls Subscriber.onNext() any number of times, followed by either Subscriber.onComplete() or Subscriber.onError().

最后调用Subscriber.onNext() 应该是调用omComplete()

10楼 Frank_Shieh 2015-04-14 12:35发表

太感谢了。膜拜了。。

9楼 liunewshine 2015-04-13 16:00发表

请问下这个和eventbus实现相似的功能吗?

8楼 sofakeer 2015-03-27 11:51发表

很感谢，学习了！！！

7楼 dywailly 2015-03-19 21:44发表

亲爱的楼主，如何在android studio里面添加rxjava的依赖呢？？

Re: Justlove_DK 2015-08-19 11:38发表

回复dywailly：你去下载一个jar包然后添加到你的lib文件里面再同步一下就可以了

Re: hi大头鬼hi 2015-03-19 22:57发表

回复dywailly：https://github.com/ReactiveX/RxAndroid
参考gradle的引用方式

6楼 sunqihui 2015-03-15 21:45发表

Reactive请翻译为反应性或反应式，Infoq上已经是这么翻译的了。而Responsive翻译为响应式，因为Responsive Web也是很流行的术语，不要搞混了！
谢谢博主！

Re: hi大头鬼hi 2015-04-14 07:57发表

回复sunqihui：好的，没注意这点

5楼 w9xhcn 2015-02-26 14:23发表

谢谢！ 期待后面的

4楼 范小八 2015-02-12 17:07发表

想实现如下功能，博主给出出主意 = =
Model类中name内容变化时，View（Activity）中Subscribers这个name的变化，同步更新Ui上的Name。
我想到的方式是
在Model中创建Obervable,在setName方法中，写入nameObservable.onNext(newName)这一行。

这样写的话，得在好多哥set函数里面加这个字段，未免不爽。博主有什么简单的写法么0-0

PS：这么做的原因是ios开发有ReactiveCocoa这个框架，类似rxjava，可以监听成员变量值变化，用习惯了且挺好用的。。。

Re: hi大头鬼hi 2015-02-15 21:14发表

回复u012410194：ReactiveCocoa是比RxJava更方便一些,RxJava一个比较好的解决你的问题的方法就是为view model创建一个compose subscription，可以参考这个项目 https://github.com/tehmou/rx-android-architecture

Re: hi大头鬼hi 2015-02-15 21:14发表

回复u012410194：ReactiveCocoa是比RxJava更方便一些,RxJava一个比较好的解决你的问题的方法就是为view model创建一个compose subscription，可以参考这个项目 https://github.com/tehmou/rx-android-architecture

3楼 hi大头鬼hi 2015-01-06 23:23发表

> 引用“xia215266092”的评论：
>
> 楼主，剩下的呢？

最近工作比较忙，要过段时间再补了

2楼 流水不腐小夏 2015-01-06 19:55发表

楼主，剩下的呢？

1楼 vicp7764 2014-12-15 15:51发表

rxjava附录里的四篇文章的翻译，之前自己蹩脚的看过了，看到lz的文章，翻译的不错，希望lz能把原文中的回复也给翻译下，其中的答复有助于理解。

Re: hi大头鬼hi 2014-12-16 22:14发表

回复vicp7764：多谢鼓励，剩下的几篇会稍后翻译