观察者模式：

定义对象间的一种一（被观察者）对多（观察者）的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

Observable

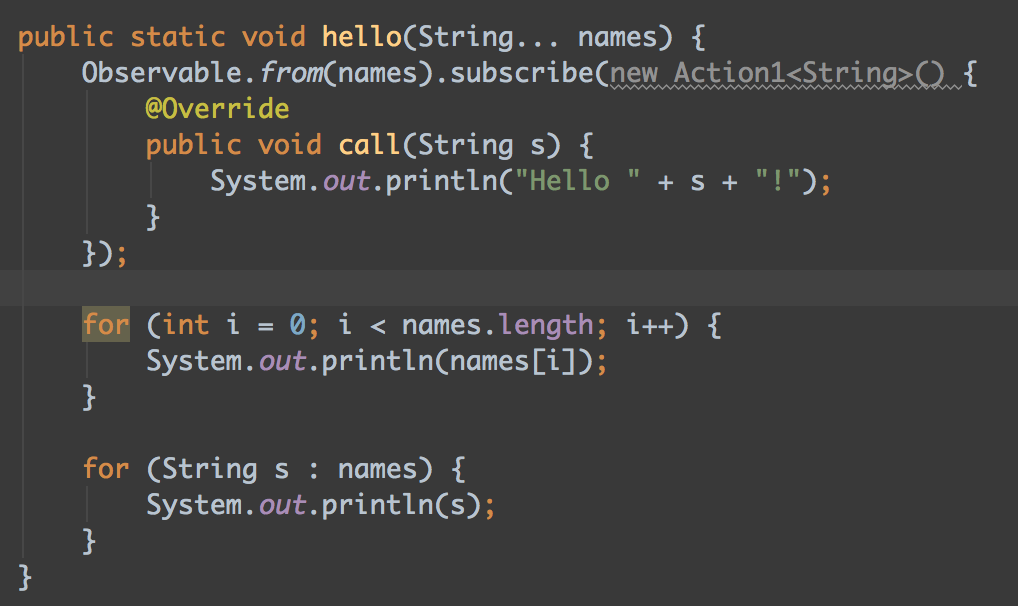
被观察者

Observer

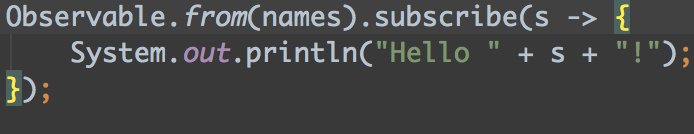
观察者

RxAndroid的观察者模式有三个回调方法（如果结合Retrofit的话里面的参数是可选的，但是onnext是必选的，举个例子，如果用户不关心结果，不关心事情否做完，只要有个onnext通知就可以，但是一般涉及到网络的话还是需要关心onerror的，做事情么，要的就是结果，但是oncompleted如果在批处理我感觉没啥卵用，后文会用代码说明），这三个onnext是一定触发的，剩下的两个有其中一个也是一定会触发的。

先来看一个简单的打印：



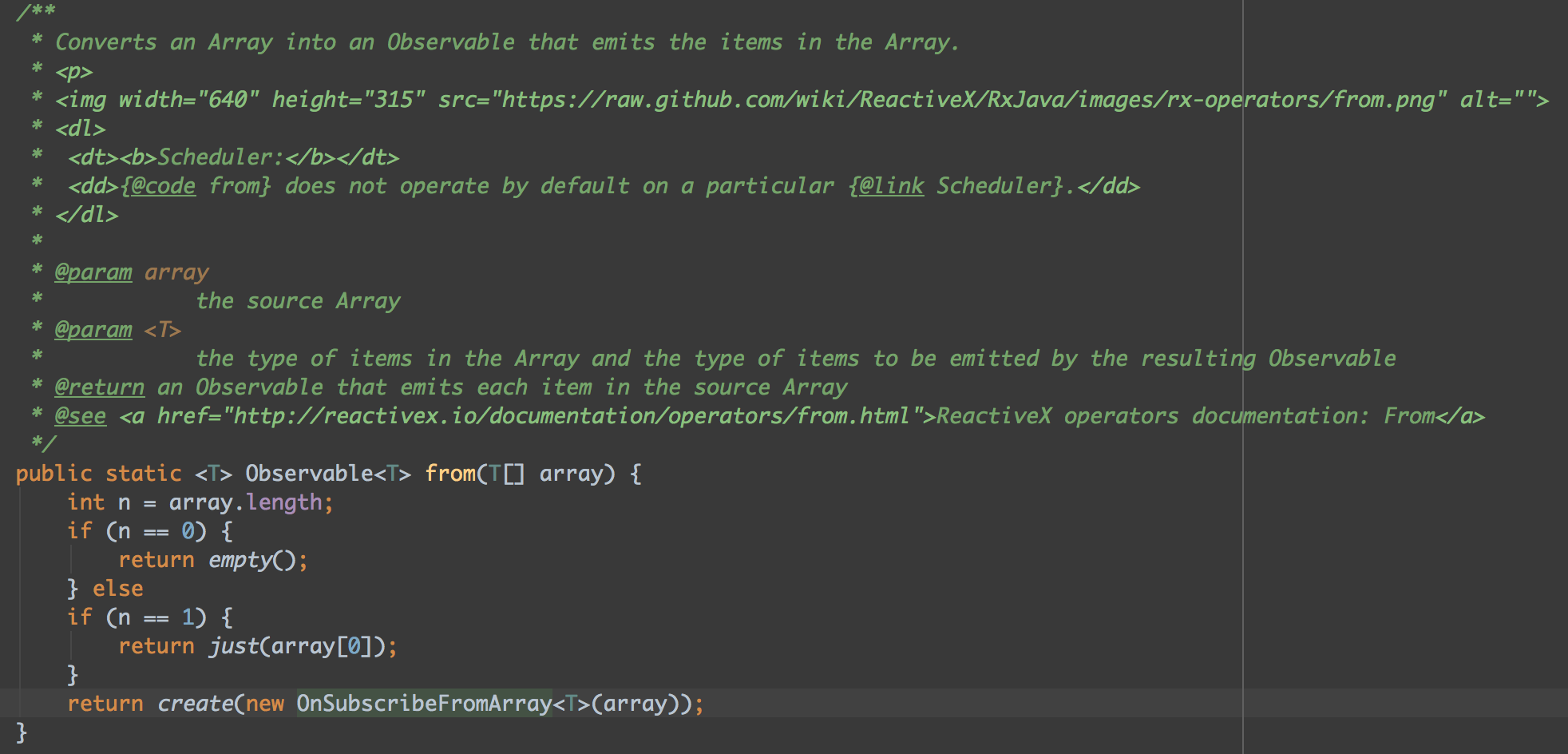
是不是感觉代码量变多了，再结合一个lambda表达式



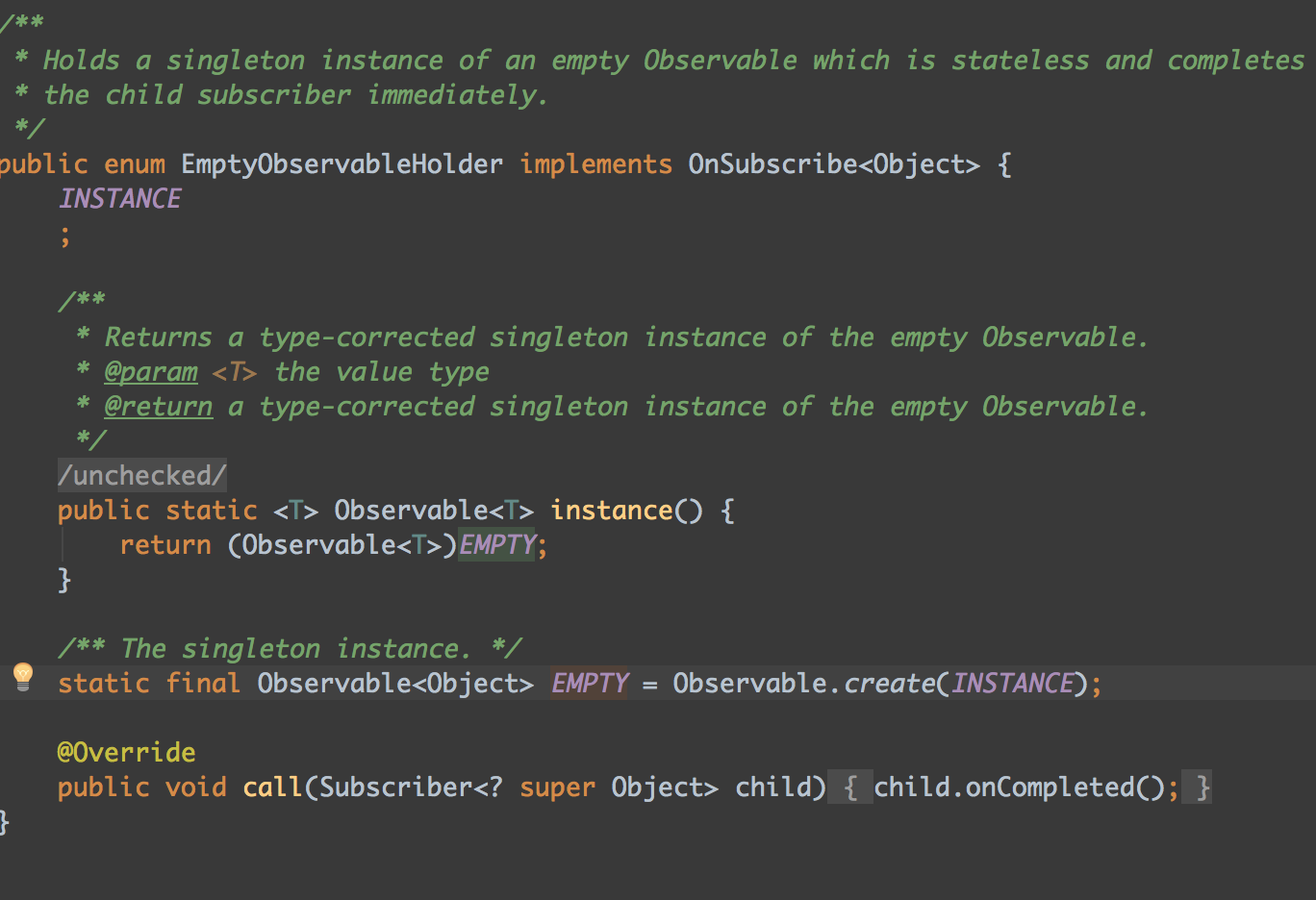
是不是又变简洁了，这个可以进行类比，as默认会进行缩进的



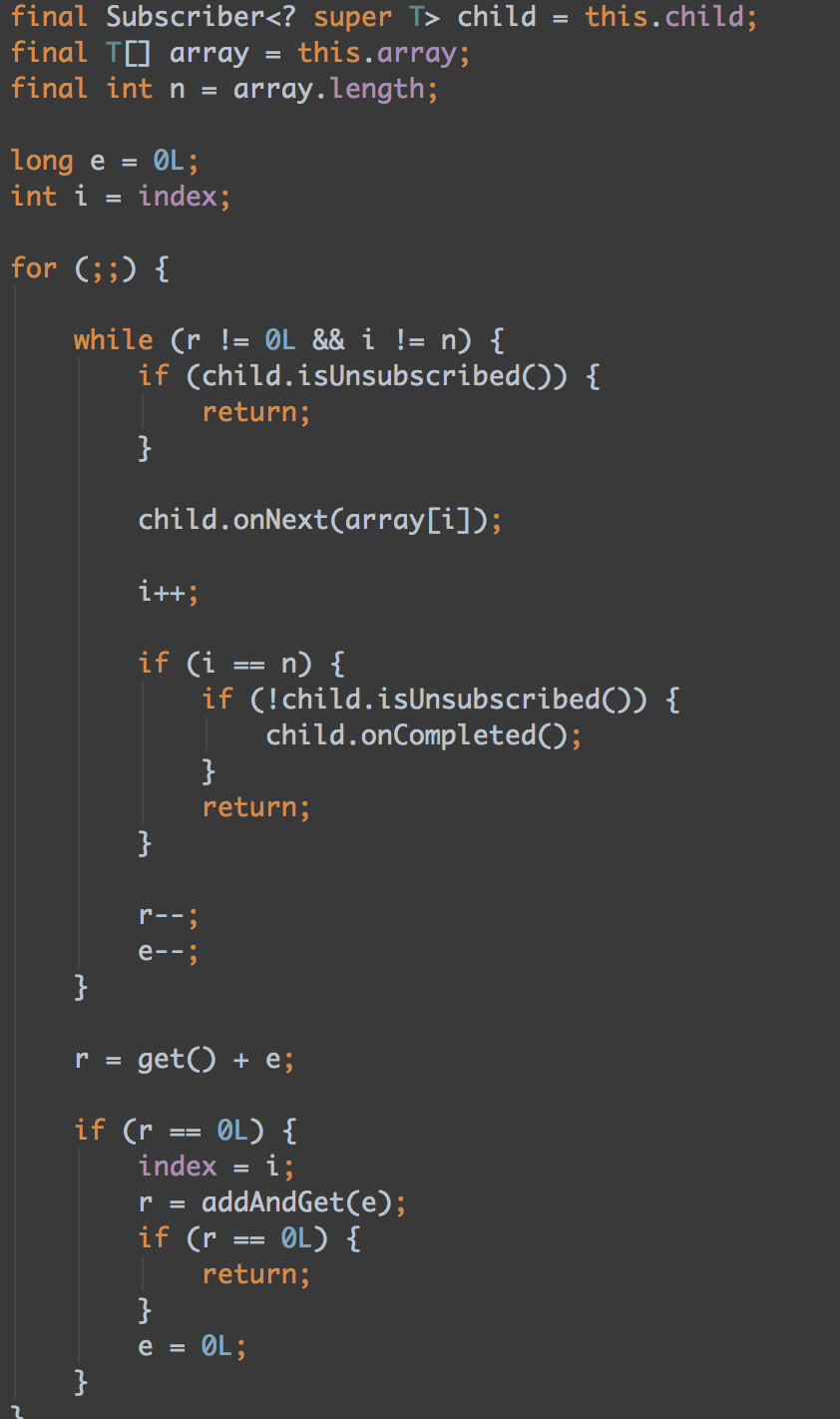
使用看了，来看一下源码，



这里的empty很有意思，我截取了一下源码，里面采用的是推荐的单例设计模式

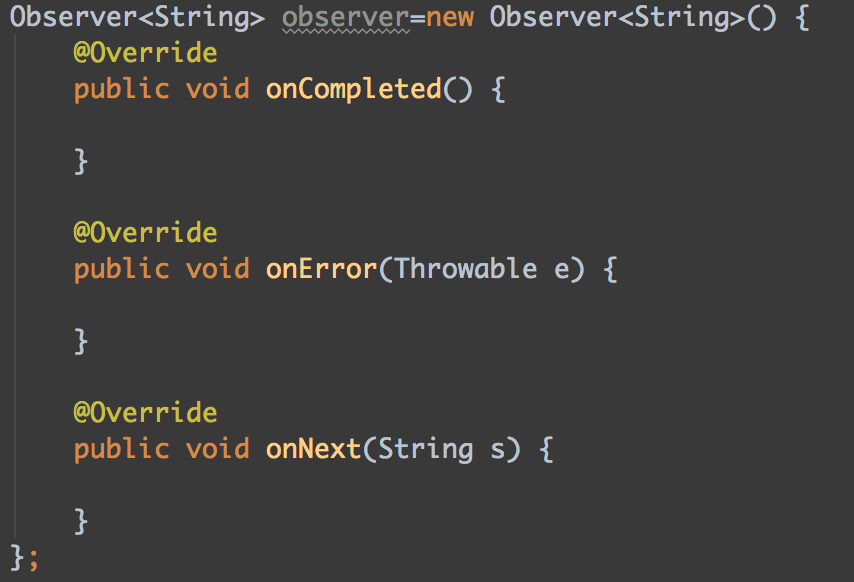


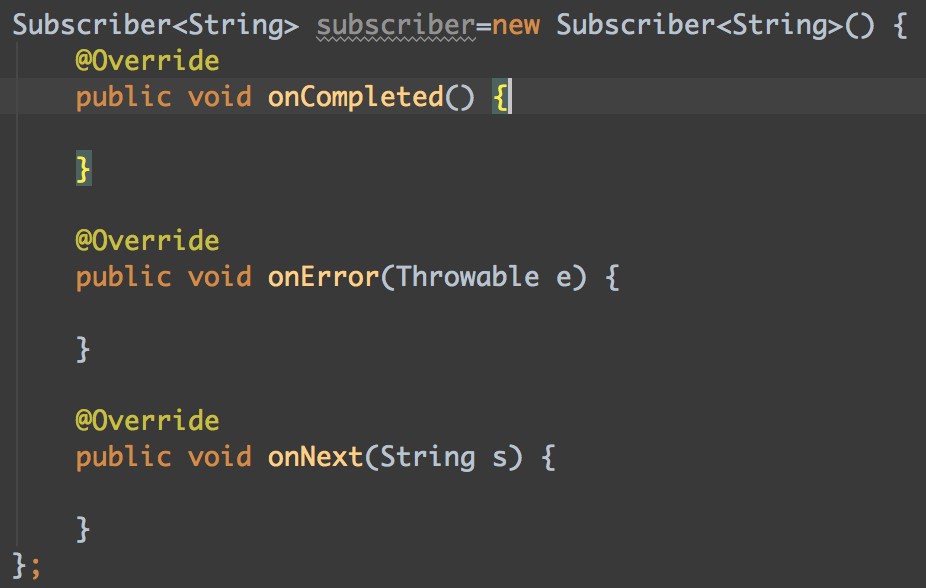
下面截取onXXXX里面的核心代码



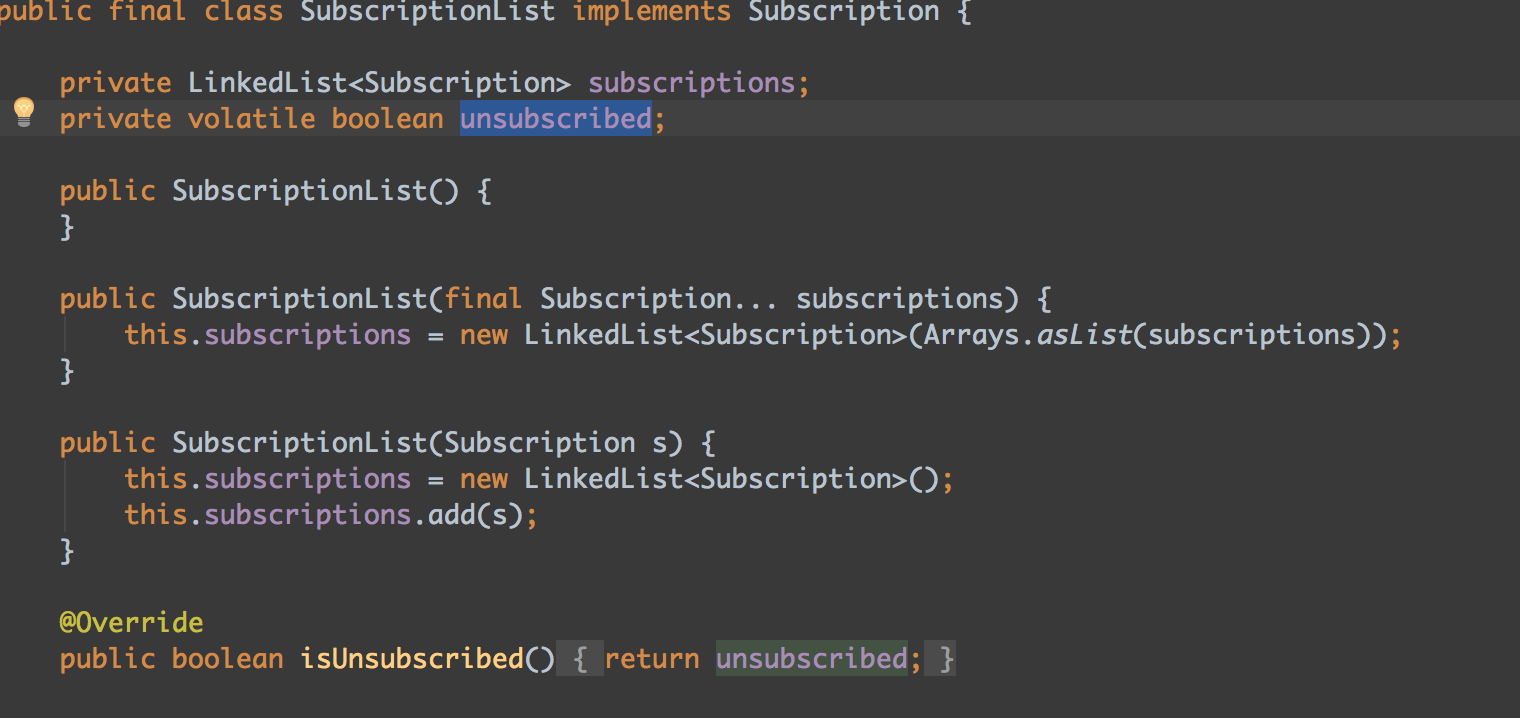
看了下稍微有点失望，其实也是用的for，但是再去具体看的话里面又结合了观察者模式还有三个回调方法，这里不细说，自己看源码。

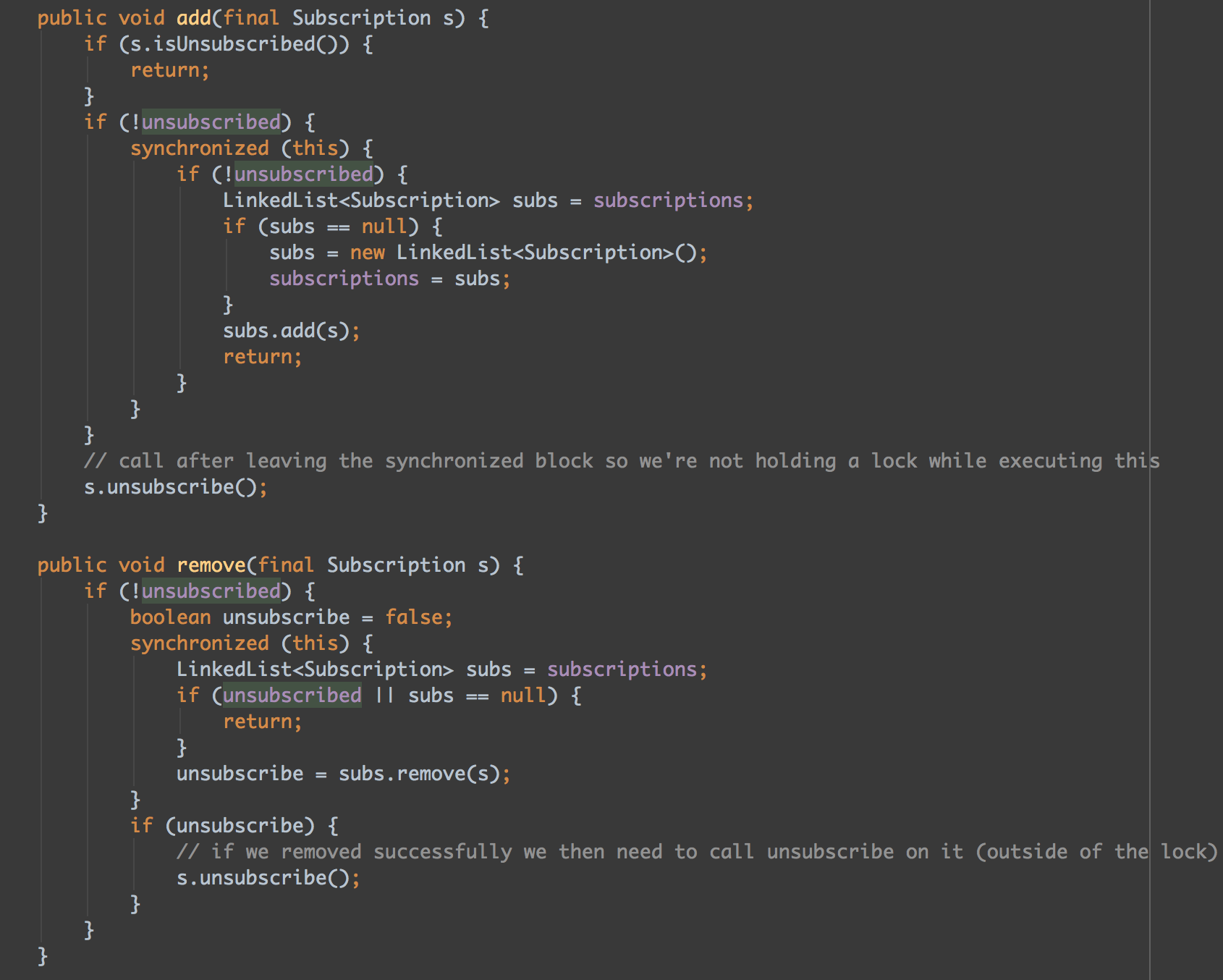
Ok,再往下看，看一下定义观察者实例和订阅者实例：



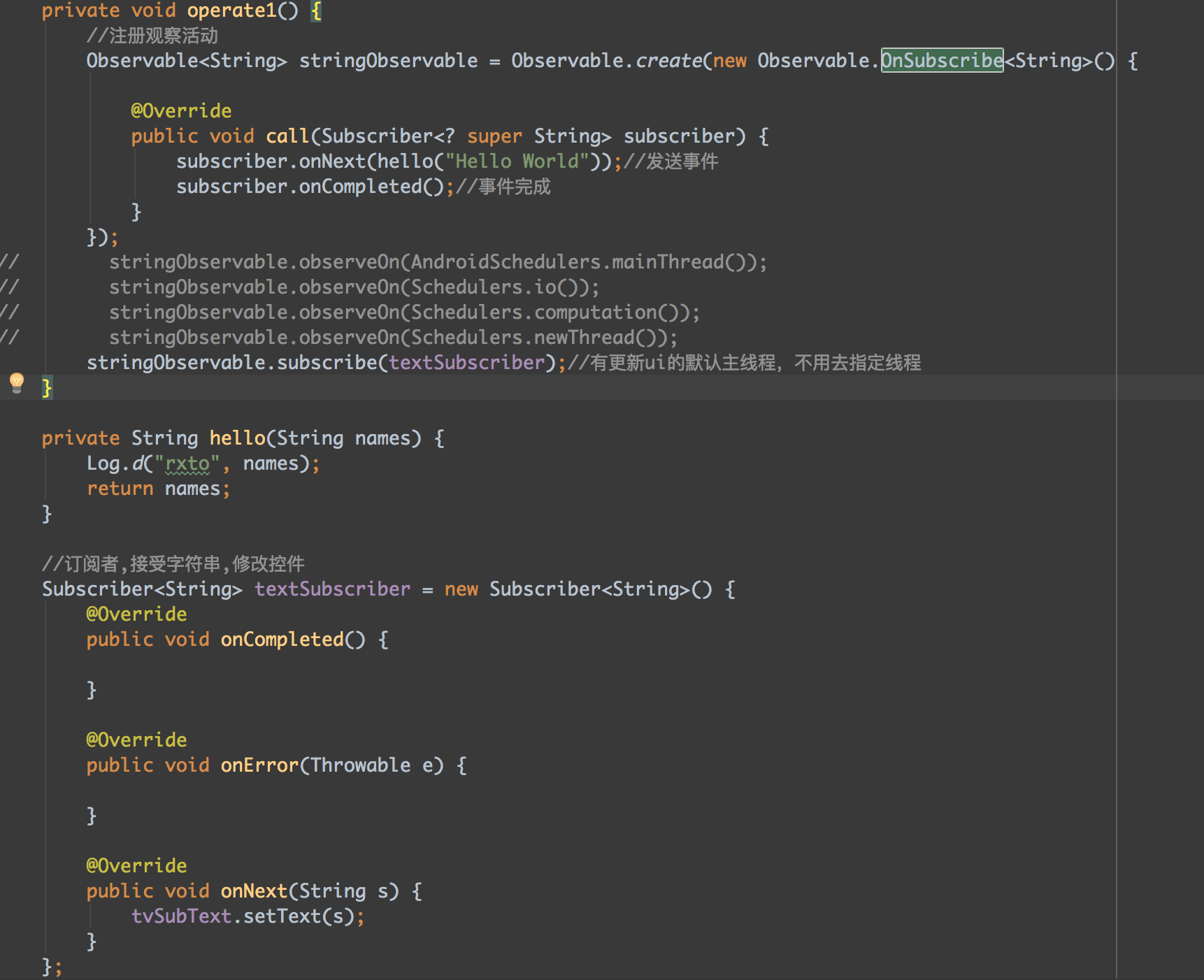


这里有段源码感觉挺不错的，截取了一段

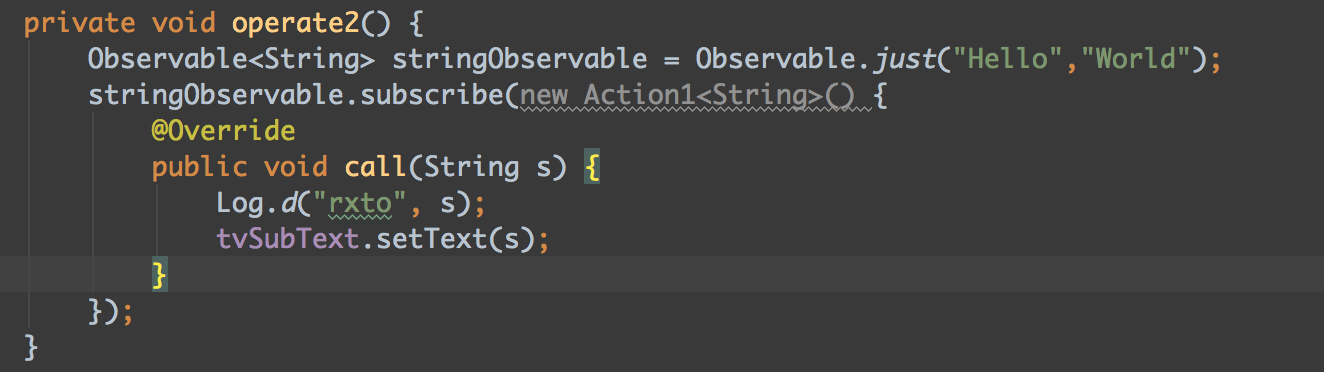




感觉到了这说的其实不少了，可是你肯定在想说这些都有毛用啊，来电实际的，先拿最普通的hello world，



我感觉看到这你心里肯定一万头草泥马路过，一行代码解决的问题搞这么多，一点都不简洁，来看简洁一点的

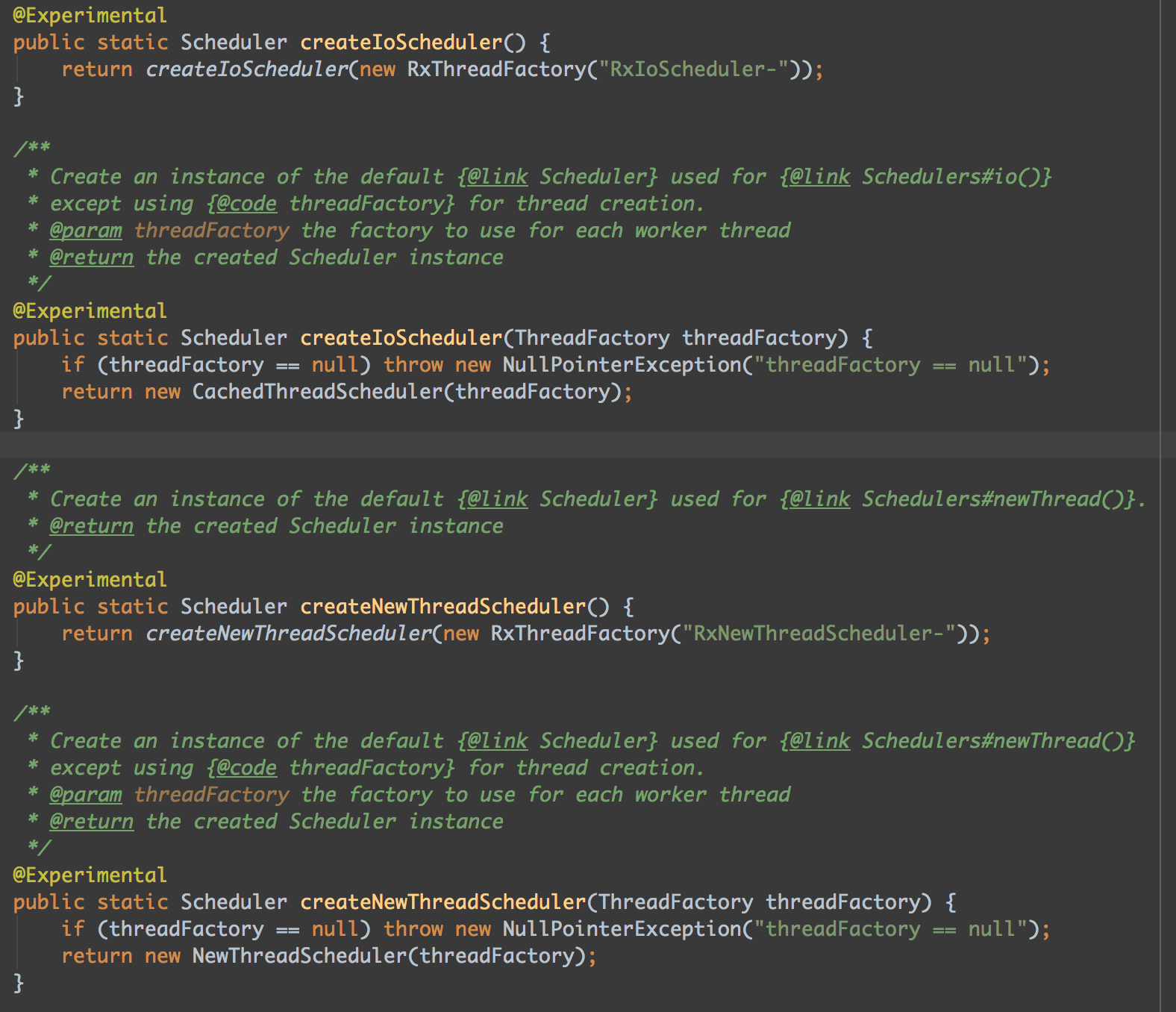




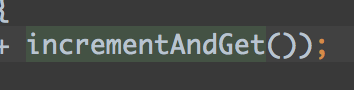
这里你肯定会问为真么只有world没有hello了，到了这里有没有感觉那么一点简洁，这里的just源码可以看一下，其实去到了from哪里，这里好多用的都是泛型，定义的时候顾虑不用太多，那连贯一下，加个过滤打印点需要的东西：

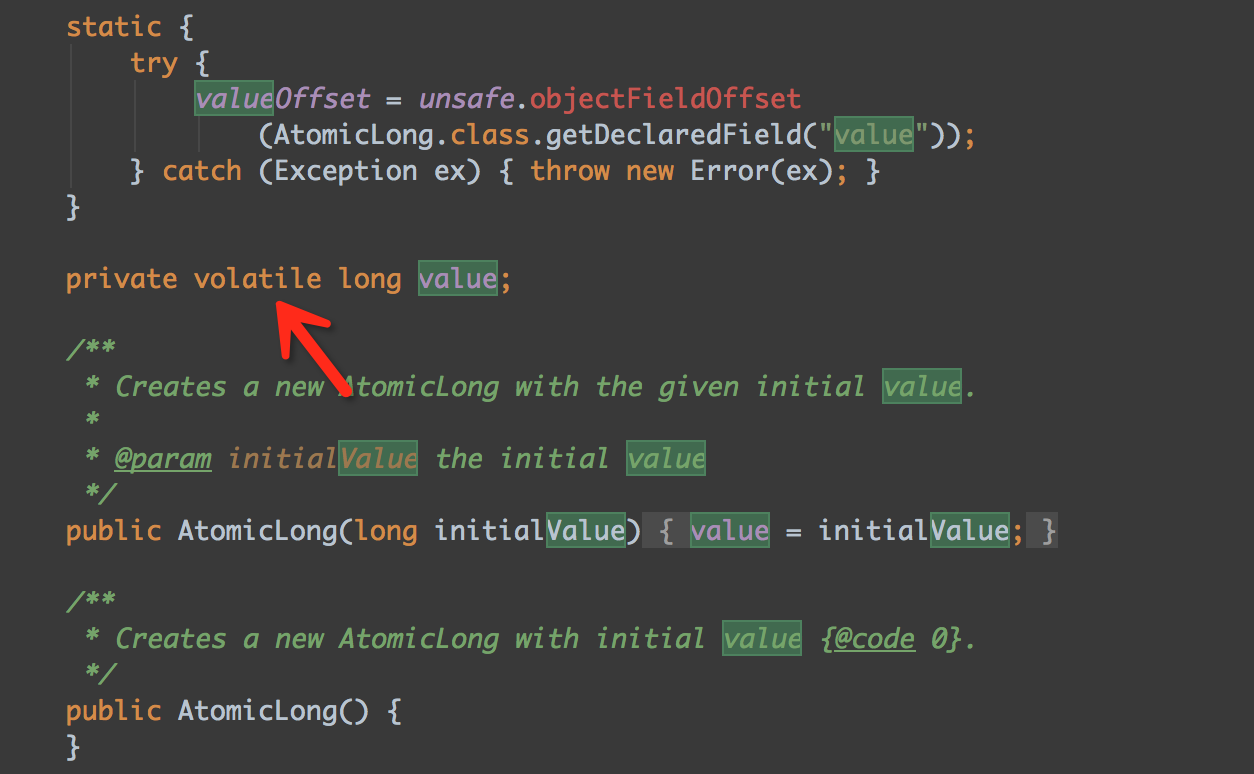


关于线程切换都有注释，下面再列举两个例子 ，看一下io，newthread的区别，因为网络请求这两个都可以，然后就勾起我的好奇心了，结果感觉貌似区别不大，至少在创建的时候





这个包是rxjava里面的，红箭头标的也不用好奇了，其实io和newthread创建过程一样，但是prefix不一样，还有也不一样，然后又去里面翻了翻，结果是这个，也是意料之中吧，但是感觉写这代码的人对性能要求也挺高的，注意箭头，不懂得google

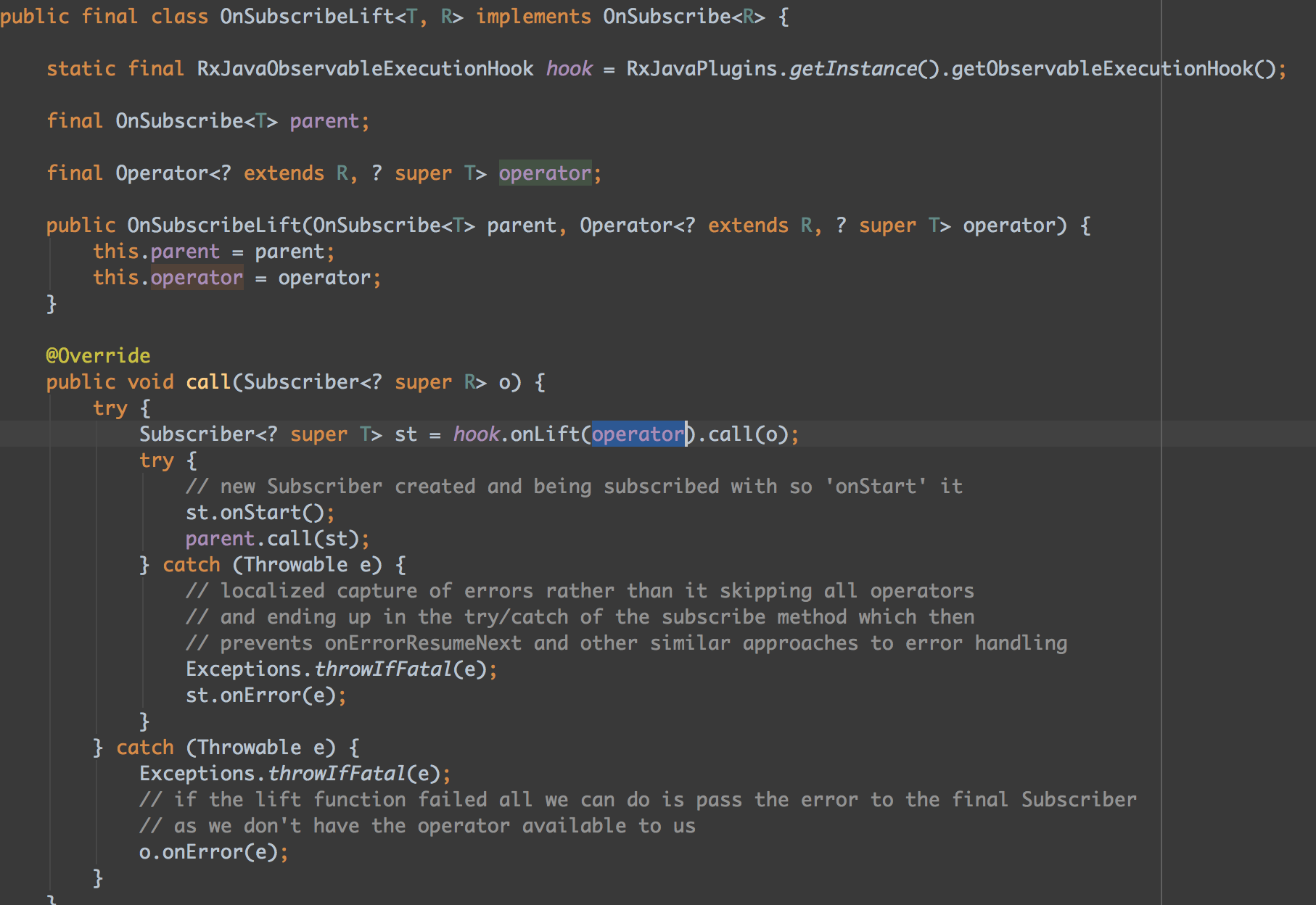


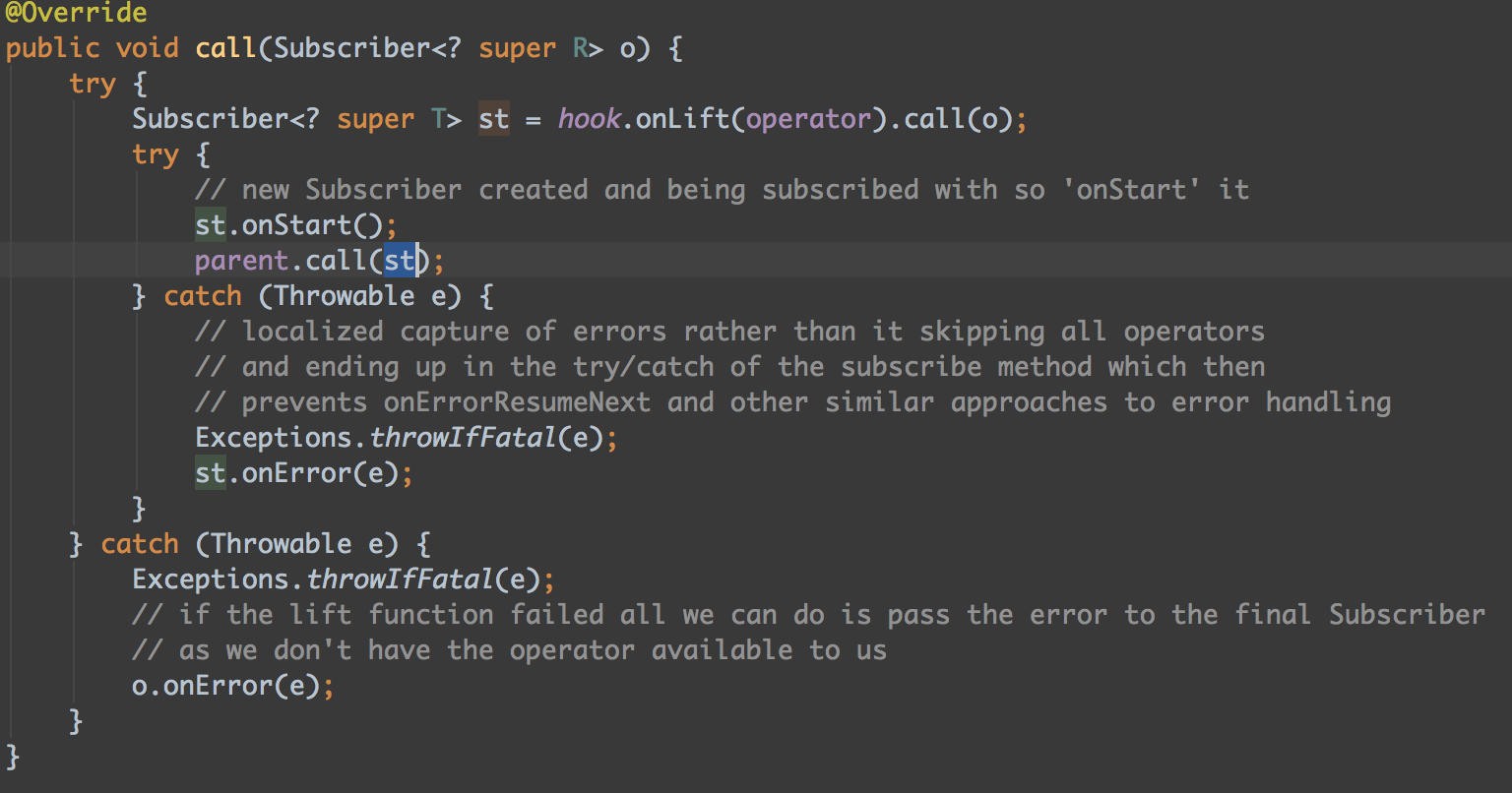




线程切换是不是很方便，还有就是感觉rxandroid在批处理方面感觉很方便，至少自己不用去写一堆循环了。

这里来看一下线程变换原理lift（）





又仔细瞅了瞅，感觉这个方法是关键，而且命名也很规范，Operator大家可以自己看源码

到此为止RxAndroid可以进行数据的处理了，再来就是加上网络请求，这里是和retrofit相结合的，rxandroid自己也可以，需要自定义callback，这里不细说了，从项目应用的角度来说一下retrofit。

简单来说retrofit就是包装了一下okhttp

retrofit感觉有大神们讲的很好了，而且官方文档说的也比较清晰详细，<http://square.github.io/retrofit/>，<http://blog.csdn.net/lmj623565791/article/details/51304204>，

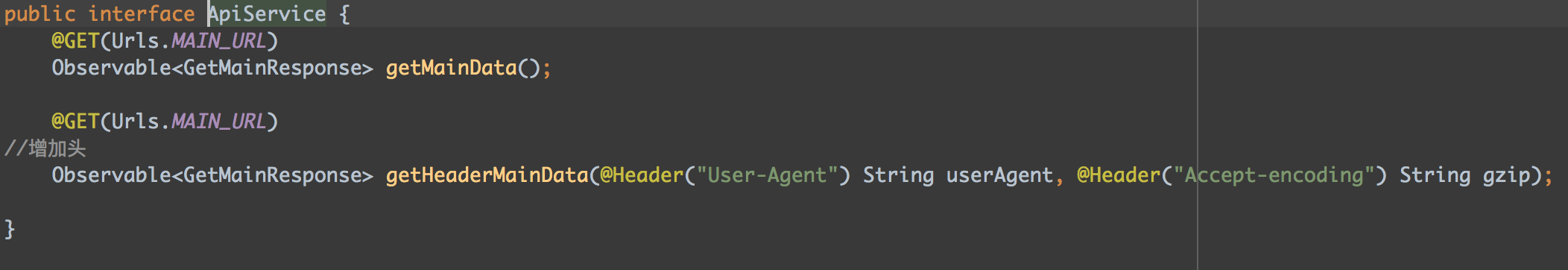
<http://mp.weixin.qq.com/s?__biz=MzA5MzI3NjE2MA==&mid=2650236203&idx=1&sn=54938620ba067eb3d6224dacb9fcaa2e&scene=23&srcid=07054QMyYdc11tfSunhoYX7m#rd>

这里就是说一下对gson不熟的，如果不进行注解的话，bean的字段需要与服务器的字段相同，并且类型需要对应



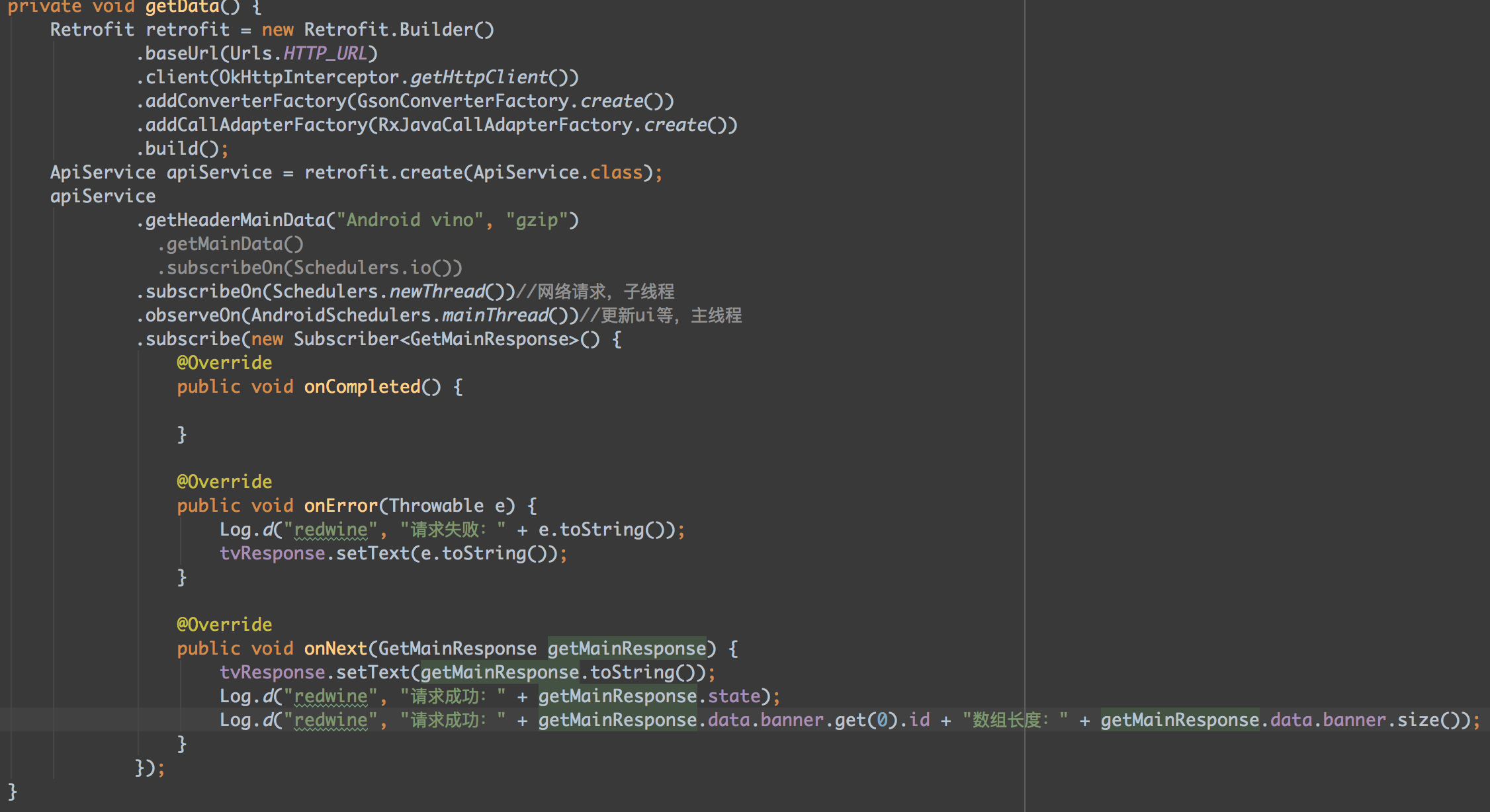
下面来说一下RxAndroid+retrofit

在这之前说一下okhttp请求拦截器的问题，一般请求的header是不会变得，但是有的是动态加的，然后我这里做了两个测试：





那现在来看一下网络请求的流程：



光看是不是就感觉很爽，一条线下来全部搞定，其实还可以再简洁一点

