

RxJava2 源码解析（一）

原创

2017年03月12日 19:42:21

标签：源码解析 / RxJava / Android / Observable / Observer



9

5651

转载请标明出处：

[http://blog.csdn.net/zxt0601/article](http://blog.csdn.net/zxt0601/article/details/61614799)本文出自：【张旭童的博客】(<http://blog.csdn.net/zxt0601>)

概述

最近事情太多了，现在公司内部变动，自己岗位的变化，以及最近决定找工作。所以博客耽误了，准备面试中，打算看一看RxJava2的源码，遂有了这篇文章。

不会对RxJava2的源码逐字逐句的，只寻找关键处，我们平时接触得到的那些代码。背压实际中接触较少，故只分析了Observable。分析的源码版本为：2.0.1

我们的目的：

1. 知道源头(Observable)是如何将数据发送出去的。
2. 知道终点(Observer)是如何接收到数据的。
3. 何时将源头和终点关联起来的
4. 知道线程调度是怎么实现的
5. 知道操作符是怎么实现的

本文先达到目的1, 2, 3。

我个人认为主要还是适配器模式的体现，我们接触的就只有Observable和Observer，其实内部有大量的中间对象在适配：将它们两联系起来，加入一些额外功能，例如考虑dispose和hook等。

从create开始

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

```
1 Observable.create(new ObservableOnSubscribe<String>() {
2     @Override
3     public void subscribe(Observer<String> e) throws Exception {
4         e.onNext("1");
5         e.onComplete();
6     }
7 }).subscribe(new Observer<String>() {
8     @Override
9     public void onSubscribe(Disposable d) {
10         Log.d(TAG, "onSubscribe() called with: d = [" + d + "]);
11     }
12
13     @Override
14     public void onNext(String value) {
15         Log.d(TAG, "onNext() called with: value = [" + value + "]);
16     }
17
18     @Override
19     public void onError(Throwable e) {
20         Log.d(TAG, "onError() called with: e = [" + e + "]);
21     }
22
23     @Override
24     public void onComplete() {
25         Log.d(TAG, "onComplete() called");
26     }
27 }
```



张旭童

博客专家

原创

51

粉丝

1431

喜欢

544



等级：博客5

访问量：337

积分：3956

排名：9573



他的最新文章

Janky frames 是如何计算出来的

2017年总结

面试必备：SparseArray源码解析

面试必备：ArrayMap源码解析

面试必备：LinkedHashMap源码(DK8)

登录

注册

文章分类

Android

程序优化和内存泄露

RecyclerView家族

设计模式

自定义LayoutManager

自定义View

展开

博主专栏



【Android】你可能知道的Support

2 篇

24665



玩转RecyclerView的面面

11 篇

129038

文章存档

27 });




拿 create来说，

```
1 public static <T> Observable<T> create(ObservableOnSubscribe<T> source) {
```

阅读全文

版权声明：本文为博主原创文章，未经博主允许不得转载。 <http://blog.csdn.net/zxt0601/article/details/61614799>

👤 目前您尚未登录，请 [登录](#) 或 [注册](#) 后进行评论

-  **Dreamfree3** 2018-03-05 19:06 回复 4楼
很清晰！
-  **he962303330** 2017-11-20 11:32 回复 3楼
楼主你真是太流弊了
-  **qq_37114521** 2017-03-15 22:58 1条回复 回复 2楼
感觉你真的是太牛逼了

查看 6 条热评

RxJava源码分析

 **u014486880** 2016年07月15日 20:36 3602

最近因为实习的原因，又好长一段时间没写博客了。今天来个较简短的博客来强势回归哈哈。上篇博客写了RN的源码分析，出乎意料的就有两个出版社联系我我要不要出书，可见RN的火热。本来也想写RN的，但因为最近实习...

RxJava2 源码解析（二）

 **zxt0601** 2017年03月20日 10:06 3015

概述 承接上一篇RxJava2 源码解析（一），本系列我们的目的：知道源头(Observable)是如何将数据发送出去的。知道终点 (Observer) 是如何接收到数据的。何时将源头和终...

一个好的前端年薪会有多少？

你认为做前端想拿到高年薪难吗？



RxJava 源码解读

 **maplejaw_** 2016年08月31日 07:50 2661

转载请注明本文出自maplejaw的博客（http://blog.csdn.net/maplejaw_）开源库地址：<https://github.com/ReactiveX/RxJava> 解读版本：...

RxJava基本流程和lift源码分析

 **lzyzsd** 2015年11月30日 11:54 24766

首先感谢 扔物线 哥哥给的配图，实在太赞了。基本结构我们先来看一段最基本的代码，分析这段代码在RxJava中是如何实现的。Observable.OnSubscribe onSubscriber1 = ...

2018年2月

2018年1月

2017年10月

2017年8月

2017年5月

2017年4月

展开

他的热门文章

【Android】详解7.0带来的新工具
ffUtil
20953

【Android】ListView、RecyclerView、ScrollView里嵌套ListView 相
17810

ConstraintLayout 属性详解 和Ch
使用
17676

【Android 仿微信通讯录 导航分
表-上】使用ItemDecoration为Re
17589

五行代码实现 炫动滑动 卡片层叠
布局，仿探探、人人影视订阅界面 仿
16511

设计模式(一) 工厂模式 五种写法总
16052

【Android】RecyclerView、List
现单选列表的优雅之路。
15505

【Android】掌握自定义LayoutM
r(二) 实现流式布局
14842

【Android】毫无耦合性，一个Ite
局搞定 item侧滑删除菜单，像IOS
13673

【Android】史上最简单，一步集
滑(删除)菜单，高仿QQ、IOS。
13236

联系我们



请扫描二维码联系：
✉ [webmaster@c](mailto:webmaster@csdn.net)
☎ 400-660-0108
🗣 QQ客服 🗣 客

关于 招聘 广告服务
©1999-2018 CSDN版权所有
京ICP证09002463号

一起来造一个RxJava，揭秘RxJava的实现原理



TellH

2017年05月10日 15:33

🔖 9270

RxJava是一个神奇的框架，用法很简单，但内部实现有点复杂，代码逻辑有点绕。我读源码时，确实有点似懂非懂的感觉。网上关于RxJava源码分析的文章，源码贴了一大堆，代码逻辑绕来绕去的，让人看得云里雾...

程序员不会英语怎么行？

老司机教你一个数学公式秒懂天下英语



我的RxJava源码解读笔记



changsimeng

2016年11月14日 12:38

🔖 3883

我的RxJava源码解读笔记 RxJava是一个用于处理异步任务的库，本篇文章我将把我在学习RxJava源码时的分析思路记载下来，一方面用来加强记忆，另一方面可以供大家参考。首先梳...

Rxjava+retrofit的简单使用demo

2016年10月24日 09:38

91KB

下载



深入解析RxJava源码（二）Observable转换原理分析

现在RxJava在Android界已经是无人不知无人不晓的技术



zhengzechuan91

2016年03月18日 16:24

🔖 1065

了，如果你还不知道RxJava是什么，推荐你去看给 Android 开发者的 RxJava 详解，而本篇博客主要是通过一些例子和源码分析以...

快速理解RxJava源码的设计理念



byamao1

2016年05月04日 10:04

🔖 1227

我在看过几篇关于RxJava源码分析的博客后，不知是我的水平有限还是源码过于博大精深，导致花了很长的时间才搞清楚其运行原理。我个人觉得应该有更好的办法来快速剖析理解，于是决定写下本文。...

RxJava源代码剖析



a1018998632

2016年05月15日 23:46

🔖 3920

转载请注明出处：何时夕的博客 很多人想学习RxJava，但是总不知道如何下手，所以我之前总结了一篇RxJava的用法。这次我举了一个例子，结合源码，深入剖析了一下，RxJava运行的机制。有爱者自...

Rxjava2源码浅析（一）



CallMeSP

2017年03月16日 16:14

🔖 686

面试的时候被问道各种框架的原理架构，也是很尴尬，自以为写的代码不少，用过的框架也不少，深入的去研究源码的还真是不多，也是给自己敲了一个警钟，今天就来尝试剖析一下Rxjava2的源码，水平有限，就先看一...

app开发公司

手机软件开发公司

百度广告



【拆轮子系列】RxJava2 源码简要分析



yxhuang2008

2017年06月20日 22:45

🔖 257

RxJava1 已经在项目中用来一年多了，最近准备转换成 RxJava2，除了去了解 2.0版本的不同，还简单的去了解了一下源码，顺便做个记录。对源码的分析，...

RxJava 源码解读分析 Scheduler



collonn

2017年05月02日 20:11

🔖 559

RxJava 源码解读分析 入门篇 Scheduler 类图分析

rxjava源码分析



firedancer0089

2018年01月09日 14:51

🔖 82

Rxjava在Android界很火，不过博主工作并不接触网络开发，没有大批量多层次的使用异步任务的机会。一直听说但是并没有使用，更不会深入分析它的机制。不过出于好奇本文会研究下源码中个人关注的点，注意...

Rxjava2 源码解析（一）



github_38075367

2017年08月10日 13:49

🔖 365

前言 Rxjava2 我们都知道是很牛逼的一个扩展的观察者模式的Scheduler，基本可以在代码中替代异步线程+Handler的开发模式，同时配合MVP设计模式，可以更好的实现业务分离。但是并不能...

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

Rxjava2源码解析超详细~~~



hbyh632 2017年09月22日 11:02 178

转载：Est 的小窝 原文地址：https://blog.codeest.moe/2017/03/25/android-rxjava2/?utm_medium=email&utm_source=gan...

程序员不会英语怎么行？

老司机教你一个数学公式秒懂天下英语



Rxjava2源码分析



wzgbgz 2018年01月25日 15:07 29

参考链接 1.<http://blog.csdn.net/zxt0601/article/details/61614799>

RxJava2 源码分析(一)



yihua0607 2017年09月08日 00:09 395

RxJava2 源码分析(二) RxJava2 源码分析(三)前言最近由于项目需要自己搭建了网络框架，采用时下非常流行的Rxjava2 + Retrofit搭建，Rxjava现在已经发展到Rxj...

16. 导入开源库 java.lang.NoClassDefFoundError: rx.plugins.RxJavaPlugins

问题：在自己项目中新导入一个开源库的Library，编译也成功运行在手机上开始出现异常，不过当时手机版本高于或等于21时却正常。报错信息：java.lang.NoClassDefFoundError:...



lTermeng 2017年05月04日 22:19 1683

导入开源库后，出现java.lang.NoClassDefFoundError: rx.plugins.RxJavaPlugi...

前言在最近的开发中，遇到较为尴尬的异常，NoClassDefFoundError，但是我称之它为异常确实不太精确，他并不是一个传统意义上的异常，与之类似的还有一个ClassNotFoundException...



mjz644638500 2017年05月25日 15:16 1129