

🔊 推荐使用Chrome、Firefox等现代浏览器浏览本站。珍爱生命，请远离IE！

RxJava操作符分类

👤 LengYue 📅 十二月 23 2015 🏷️ Android RxJava

ReactiveX的每种编程语言的实现都实现了一组操作符的集合。不同的实现之间有很多重叠的部分，也有一些操作符只存在特定的实现中。每种实现都倾向于用那种编程语

LengYue's Blog 首页 Android 岁月如歌 🔍

有一个决策树

用于帮助你根据具体的场景选择合适的操作符。最后有一个语言特定实现的按字母排序的操作符列表。

如果你想实现你自己的操作符，可以参考这里：实现自定义操作符

创建操作

用于创建Observable的操作符

- Create – 通过调用观察者的方法从头创建一个Observable
- Defer – 在观察者订阅之前不创建这个Observable，为每一个观察者创建一个新的Observable
- Empty/Never/Throw – 创建行为受限的特殊Observable
- From – 将其它的对象或数据结构转换为Observable
- Interval – 创建一个定时发射整数序列的Observable
- Just – 将对象或者对象集合转换为一个会发射这些对象的Observable
- Range – 创建发射指定范围的整数序列的Observable
- Repeat – 创建重复发射特定的数据或数据序列的Observable
- Start – 创建发射一个函数的返回值的Observable
- Timer – 创建在一个指定的延迟之后发射单个数据的Observable

变换操作

这些操作符可用于对Observable发射的数据进行变换，详细解释可以看每个操作符的文档

- uffer – 缓存，可以简单的理解为缓存，它定期从Observable收集数据到一个集合，然后把这些数据集合打包发射，而不是一次发射一个
- latMap – 扁平映射，将Observable发射的数据变换为Observables集合，然后将这些Observable发射的数据平坦化的放进一个单独的Observable，可以认为是一个将嵌套的数

据结构展开的过程。

groupBy — 分组，将原来的Observable分拆为Observable集合，将原始Observable发射的数据按Key分组，每一个Observable发射一组不同的数据

map — 映射，通过对序列的每一项都应用一个函数变换Observable发射的数据，实质是对序列中的每一项执行一个函数，函数的参数就是这个数据项

scan — 扫描，对Observable发射的每一项数据应用一个函数，然后按顺序依次发射这些值

window — 窗口，定期将来自Observable的数据分拆成一些Observable窗口，然后发射这些窗口，而不是每次发射一项。类似于Buffer，但Buffer发射的是数据，Window发射的是Observable，每一个Observable发射原始Observable的数据的一个子集

过滤操作



这些操作符用于从Observable发射的数据中进行选择

Debounce — 只有在空闲了一段时间后才发射数据，通俗的说，就是如果一段时间没有操作，就执行一次操作

Distinct — 去重，过滤掉重复数据项

ElementAt — 取值，取特定位置的数据项

Filter — 过滤，过滤掉没有通过谓词测试的数据项，只发射通过测试的

First — 首项，只发射满足条件的第一条数据

IgnoreElements — 忽略所有的数据，只保留终止通知(onError或onCompleted)

Last — 末项，只发射最后一条数据

Sample — 取样，定期发射最新的数据，等于是数据抽样，有的实现里叫ThrottleFirst

Skip — 跳过前面的若干项数据

SkipLast — 跳过后面的若干项数据

Take — 只保留前面的若干项数据

TakeLast — 只保留后面的若干项数据

组合操作

组合操作符用于将多个Observable组合成一个单一的Observable

And/Then/When — 通过模式(And条件)和计划(Then次序)组合两个或多个Observable发射的数据集

CombineLatest — 当两个Observables中的任何一个发射了一个数据时，通过一个指定的函数组合每个Observable发射的最新数据（一共两个数据），然后发射这个函数的结果

Join — 无论何时，如果一个Observable发射了一个数据项，只要在另一个Observable发射的数据项定义的时间窗口内，就将两个Observable发射的数据合并发射

Merge — 将两个Observable发射的数据组合成一个

StartWith — 在发射原来的Observable的数据序列之前，先发射一个指定的数据序列或数据项

Switch — 将一个发射Observable序列的Observable转换为这样一个Observable：它逐个发射那些Observable最近发射的数据

Zip — 打包，使用一个指定的函数将多个Observable发射的数据组合在一起，然后将这个函数的结果作为单项数据发射

错误处理

这些操作符用于从错误通知中恢复

Catch — 捕获，继续序列操作，将错误替换为正常的的数据，从onError通知中恢复

Retry — 重试，如果Observable发射了一个错误通知，重新订阅它，期待它正常终止

辅助操作

一组用于处理Observable的操作符

Delay — 延迟一段时间发射结果数据

Do — 注册一个动作占用一些Observable的生命周期事件，相当于Mock某个操作

Materialize/Dematerialize — 将发射的数据和通知都当做数据发射，或者反过来

ObserveOn — 指定观察者观察Observable的调度程序（工作线程）

Serialize — 强制Observable按次序发射数据并且功能是有效的

Subscribe — 收到Observable发射的数据和通知后执行的操作

SubscribeOn — 指定Observable应该在哪个调度程序上执行

TimeInterval — 将一个Observable转换为发射两个数据之间所耗费时间的Observable

Timeout — 添加超时机制，如果过了指定的一段时间没有发射数据，就发射一个错误通知

Timestamp — 给Observable发射的每个数据项添加一个时间戳

Using — 创建一个只在Observable的生命周期内存在的一次性资源

条件和布尔操作

这些操作符可用于单个或多个数据项，也可用于Observable

All — 判断Observable发射的所有的数据项是否都满足某个条件

Amb — 给定多个Observable，只让第一个发射数据的Observable发射全部数据

Contains — 判断Observable是否会发射一个指定的数据项

DefaultIfEmpty — 发射来自原始Observable的数据，如果原始Observable没有发射数据，就发射一个默认数据

SequenceEqual — 判断两个Observable是否按相同的数据序列

SkipUntil — 丢弃原始Observable发射的数据，直到第二个Observable发射了一个数据，然后发射原始Observable的剩余数据

SkipWhile — 丢弃原始Observable发射的数据，直到一个特定的条件为假，然后发射原始Observable剩余的数据

TakeUntil — 发射来自原始Observable的数据，直到第二个Observable发射了一个数据或一个通知

TakeWhile — 发射原始Observable的数据，直到一个特定的条件为真，然后跳过剩余的数据

算术和聚合操作

这些操作符可用于整个数据序列

Average — 计算Observable发射的数据序列的平均值，然后发射这个结果

Concat — 不交错的连接多个Observable的数据

- Count – 计算Observable发射的数据个数，然后发射这个结果
- Max – 计算并发射数据序列的最大值
- Min – 计算并发射数据序列的最小值
- Reduce – 按顺序对数据序列的每一个应用某个函数，然后返回这个值
- Sum – 计算并发射数据序列的和

连接操作

一些有精确可控的订阅行为的特殊Observable

- Connect – 指示一个可连接的Observable开始发射数据给订阅者
- Publish – 将一个普通的Observable转换为可连接的
- RefCount – 使一个可连接的Observable表现得像一个普通的Observable
- Replay – 确保所有的观察者收到同样的数据序列，即使他们在Observable开始发射数据之后才订阅

转换操作

- To – 将Observable转换为其它的对象或数据结构
- Blocking 阻塞Observable的操作符

操作符决策树

几种主要的需求

- 直接创建一个Observable（创建操作）
- 组合多个Observable（组合操作）
- 对Observable发射的数据执行变换操作（变换操作）
- 从Observable发射的数据中取特定的值（过滤操作）
- 转发Observable的部分值（条件/布尔/过滤操作）
- 对Observable发射的数据序列求值（算术/聚合操作）

转载请注明：[LengYue's Blog](#) » [RxJava操作符分类](#)


分享到：

0条评论

最新 最早 最热

还没有评论，沙发等你来抢

社交帐号登录: 微信 微博 QQ 人人 [更多»](#)



说点什么吧...

发布



LengYue's Blog - 琥珀之月,空谷之兰,皆是孤芳,既是孤芳,但求自赏便罢!



琥珀之月,空谷之兰,皆是孤芳,既是孤芳,但求自赏便罢!

📌 标签云

- 反射
- 动态代理
- Java
- Bitmap
- 原理
- 热修复
- 急速打包
- Android M
- PopupWindow
- 网易云音乐
- Emoji
- DrawableLeft
- MySQL
- 开发资源
- RelativeLayout
- RxJava
- Glide
- Fresco
- Imageloader
- Picasso
- Menu
- android.os.BinderProxy
- LinearLayout
- ShareSDK
- 银行卡号
- StackOverflowError
- Activity
- super-wingpanel
- sublime
- Elemetary OS
- 应用市场
- Gradle
- 打包
- merge
- ASimpleCache
- Launch Activity
- android:windowIsTranslucent
- Volley
- Intent
- 缩放、旋转
- ThumbnailUtils
- DownloadManager
- 汉字转换成拼音
- ScrollView
- sqlite
- IDEA
- TextView
- Fastjson
- RoundedImageView
- 超链接
- NotificationCompat
- receiver
- AndroidManifest
- Padding
- emacs
- GUN
- aapt
- 苏菲的世界
- DatePickerDialog
- Linux
- BaiduAPI
- tools
- Ubuntu
- 图库
- Git
- SVN
- MongoDB
- Fragment
- 百度地图
- ViewPager
- GPS
- ListView
- 线程
- Timer
- 数据流
- ADB
- NFC
- Android
- 软件开发
- 岁月如歌
- 系统简介

友情链接

[Ghost中国](#)

[CSDN](#)

本站基于 Ghost 采用 ghost-matery主题搭建. 陕ICP备2332333号

