https://zhuanlan.zhihu.com/p/29130168

RxJS 中的 Subjects 经常被误解。因为它们允许你命令式地向 Observable 流中推送值，当人们不太清楚如何将某个东西变成 Observable 时，他们倾向于滥用 Subjects 。此模式看起来有点像这样...

const subject = new Subject();

button.addEventListener('click', () => subject.next('click'));

subject.subscribe(x => console.log(x));

虽然这对于 RxJS 新手 (对于这个阶段来说太正常不过了) 很有帮助，但这不是以 “Rx 的方式”在处理问题。理想的是在 Observable 中包装事件注册，既可以监听事件，又可以取消事件监听。看起来像这样:

// 这样好些了，但请使用 Observable.fromEvent(button, 'click') 来替代

const clicks = new Observable(observer => {

const handler = (e) => observer.next(e);

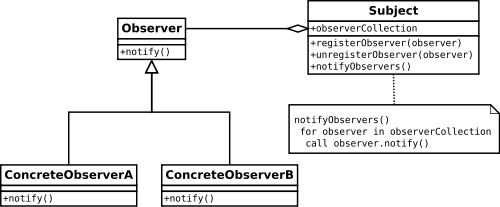
button.addEventListener('click', handler);

return () => button.removeEventListener('click', handler);});

为什么展示这个跟 Subjects 没半点关系的示例？好吧，一是它展示了为什么不总是需要使用 Subject，另外这有个隐藏的 subject... (某种程度上可以说是 subject )。这里要注意的一点是，Observable 是通过 addEventListener 来包装按钮的处理函数的注册，而 addEventListener 本身就是一个 subject 。…至少根据 [“Gang Of Four” 的观察者模式](https://link.zhihu.com/?target=https://en.wikipedia.org/wiki/Design_Patterns" \t "https://zhuanlan.zhihu.com/p/_blank)来说是这样的。

**观察者模式**

你可能知道，RxJS 主要是关于 Observables 和 Observers 的，但它也是 Subjects 相关的。然而在 Gof 的 [设计模式](https://link.zhihu.com/?target=https://en.wikipedia.org/wiki/Design_Patterns" \t "https://zhuanlan.zhihu.com/p/_blank)中是找不到 observables 的，Subjects 和 Observers 是 [观察者模式](https://link.zhihu.com/?target=https://en.wikipedia.org/wiki/Observer_pattern" \t "https://zhuanlan.zhihu.com/p/_blank)中的根本。



模式本身很简单。Observers 是具有通知方法的类，Subject 也是类，它具有向内部观察者列表添加或删除观察者的方法和通知观察者列表的方法。

RxJS 中的 Subjects 并没有太差区别。当使用 observer 对 Rx Subject 调用 `subscribe` 时，Subject 会将该 observer 添加到内部的观察者列表中。同样的，如果使用一到三个函数来调用 `subscribe`，Subject 会将它们包装成一个 observer，然后添加到观察者列表中。当调用 Subject 的 `next(value)` 时，它会遍历观察者列表并将 `value` 传递给 `next` 方法。对于 `error` 和 `complete` 也是同样的。要想从 subject 的观察者列表中移除 observer，只需简单调用 subscription 的 `unsubscribe` 方法即可，subscription 是将 observer 添加到观察者列表中时返回的。

const subject = new Subject();

const sub1 = subject.subscribe(observer1); // 将 observer1 添加到观察者列表

const sub2 = subject.subscribe(observer2); // 将 observer2 添加到观察者列表

subject.next('hi there'); // 使用 "hi there" 来通知列表中的所有观察者

sub1.unsubscribe(); // 将 observer1 从观察者列表中移除

## **Subject 相比较于 Observable**

实际上，RxJS 中的 Subjects 不同于 GoF 观察者模式中的 Subjects，但它们的 API 是 Observable 的鸭子类型。实际上，在 RxJS 中 Subjects 更甚，它们继承自 Observable 。优点是所有 Subjects 都具有与 Observable 相同的操作符和方法。

大概 Subject 和 Observable 之间一个很重要的区别就是 Subject 是有状态的，它维护观察者列表。另一方面，[Observable 真的只是一个函数](https://link.zhihu.com/?target=https://medium.com/@benlesh/learning-observable-by-building-observable-d5da57405d87" \t "https://zhuanlan.zhihu.com/p/_blank)，它建立了观察本身。

虽然 Subjects 是 Observables，但 Subjects 还实现了 Observer 接口。也就是说，它们拥有 `next`、`error` 和 `complete` 方法。这些方法用来通知 subject 内部观察者列表中的 observers 。这意味着 subject 可以用作订阅任何 observable 的 observer 。