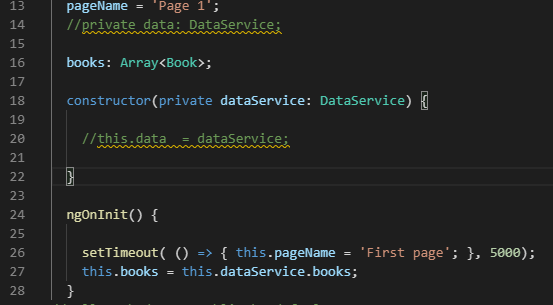
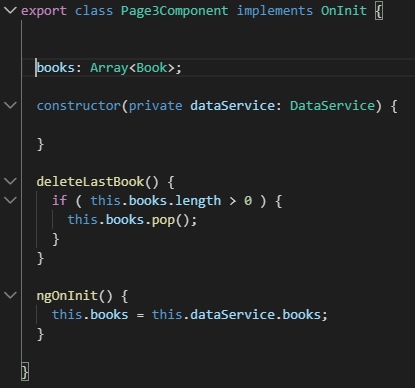
**Observer Pattern**

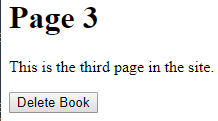
Used throughout Angular:

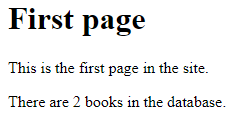
Recoded the dataService example







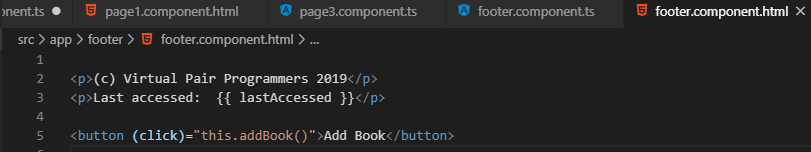




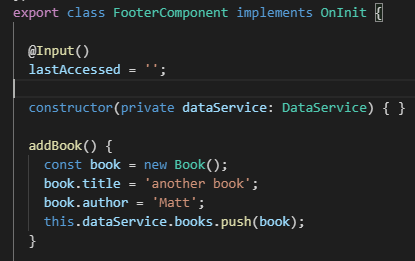
**It still works!**

**Add a book in the Footer component:**

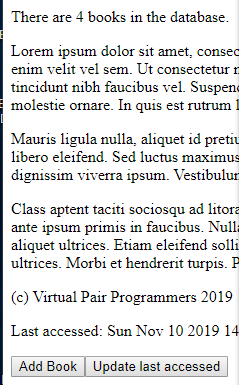
**Now, in the footer.component.html add the button with the method firing off the click event:**



**In the footer.component.ts file add the method**

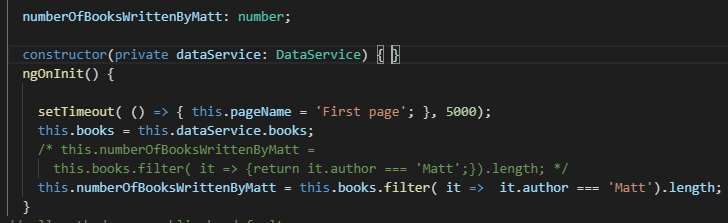


**Selected Add Book and the books change to 4**

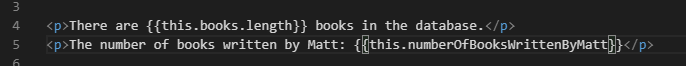


**Now, lets find how many books are written by Matt (same author)**

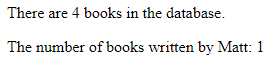
**Use a lambda function to find the number books:**



**The long syntax and the shortened version are showed**



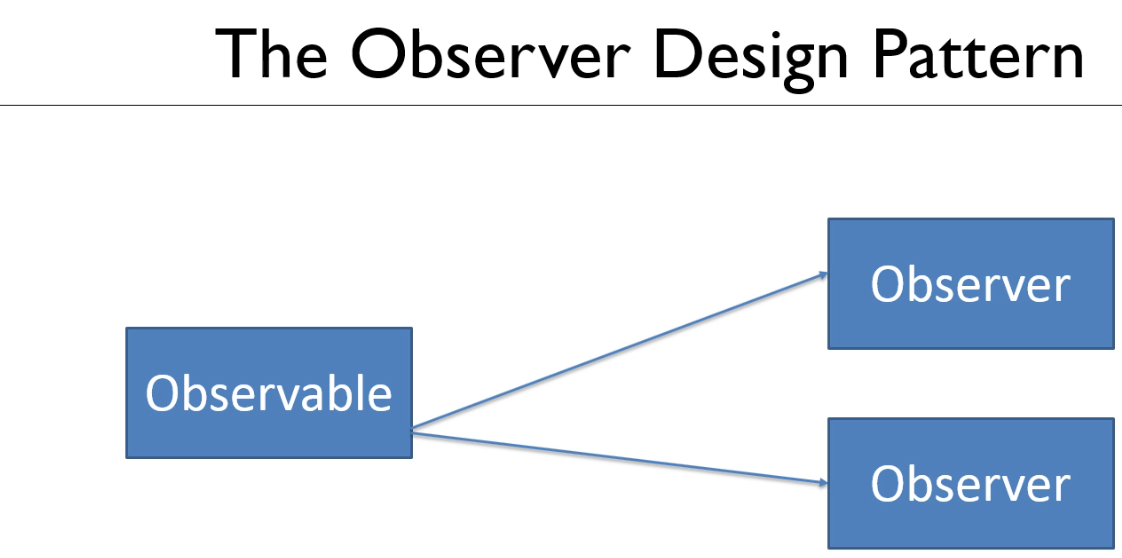
**Testing shows that the length does not get updated dynamically**



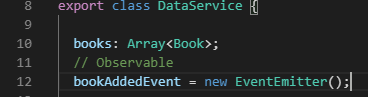
**Matt should have changed from 1 to 2 when we added the fourth book.**

**Now let us introduce the Observer Pattern**

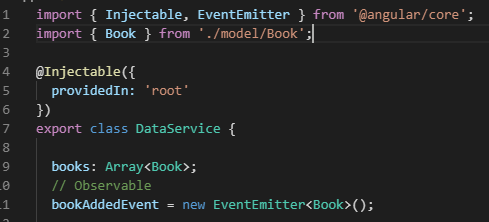
**When an Observable triggers an event the listener Observers respond**



**Let us create an Observable in the Injectable DataService Service:**

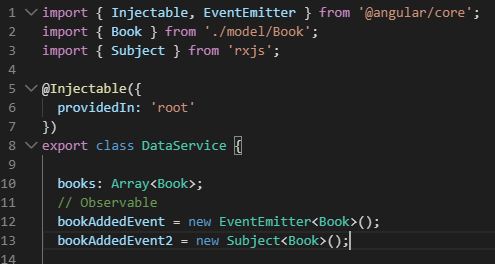


**If we want to send book data along with the event make the generic type <Book>**



**Notice that the Import Injectable has the EventEmitter as well**

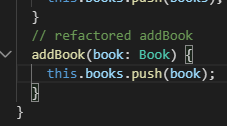
**You can also use the rxjs or react version:**



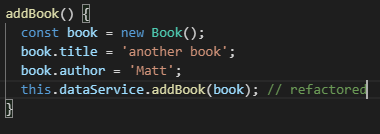
**Let us now trigger the event:**

**First refactor addBook()**

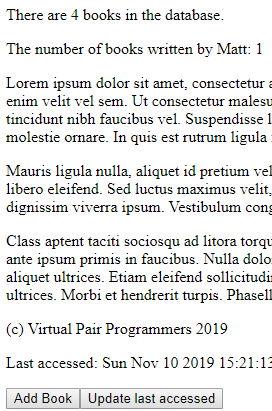
**Added to dataService**



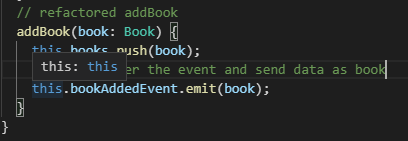
**Modified in the footer.component.ts**



**Testing**

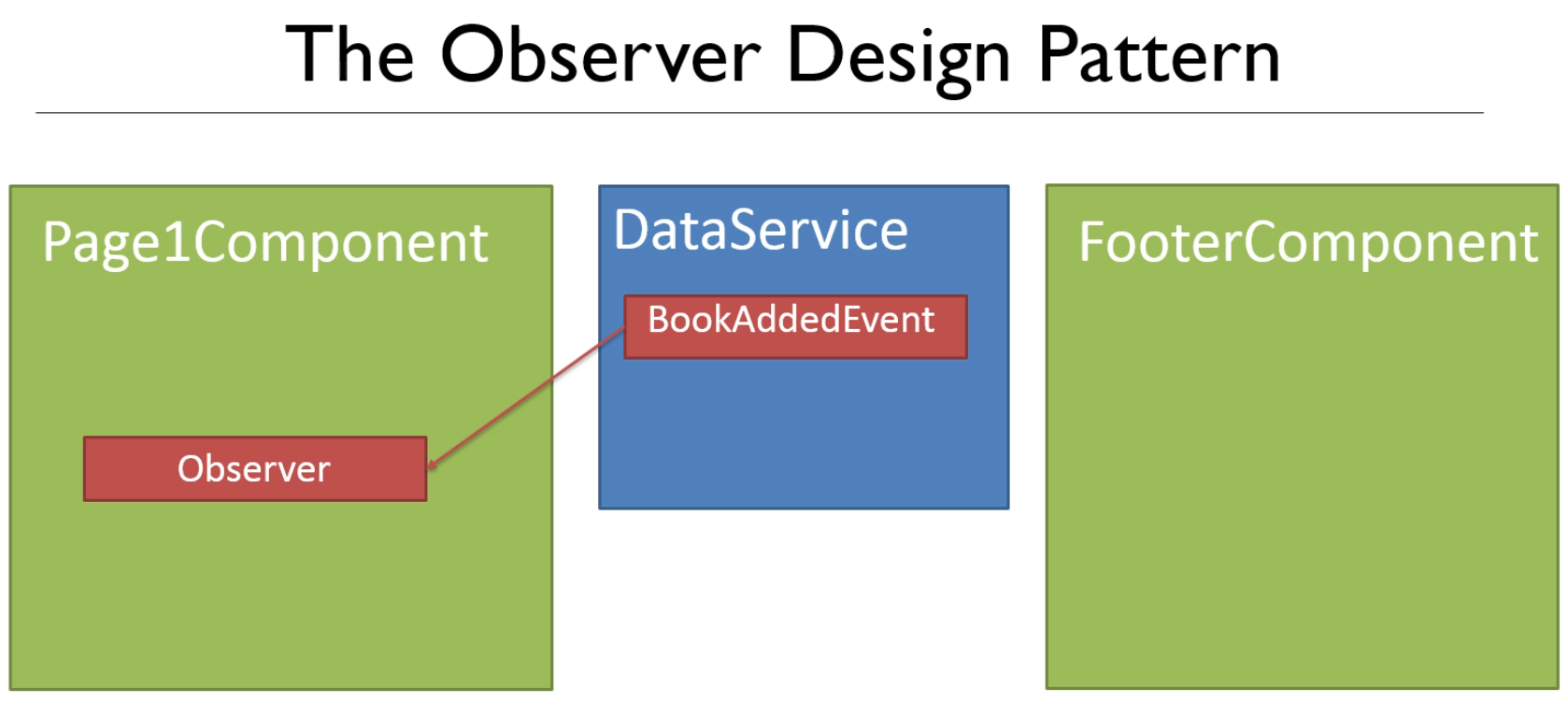


**Now, trigger the event and send the data to listeners**

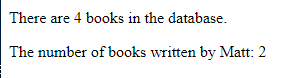


**Adding a subscription to the Observer:**



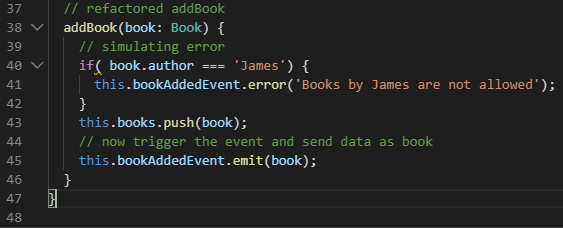


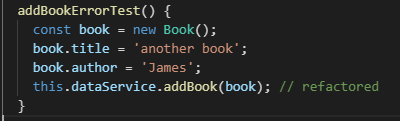
**Let us test this:**



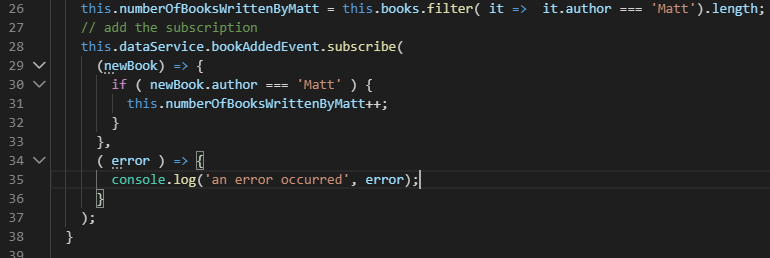
**It works!!**

**Dealing with Errors:**

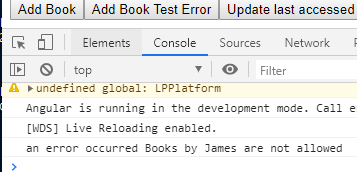








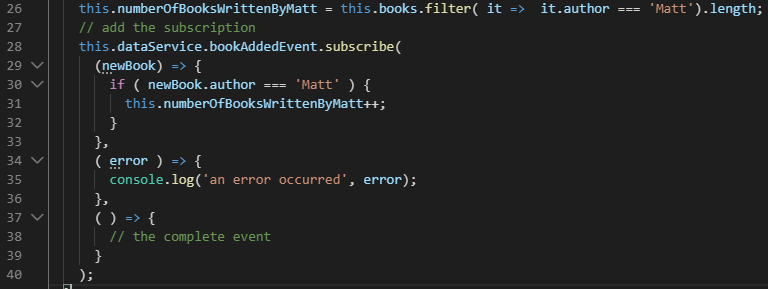
**Test this**



**It works!**

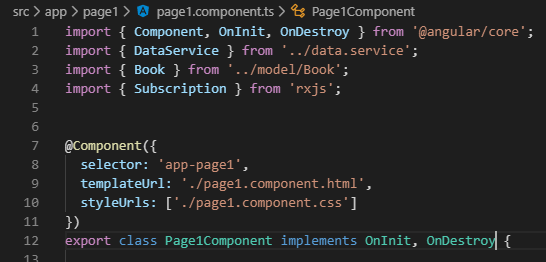
**Subscription ends on error so, Matt does not update**

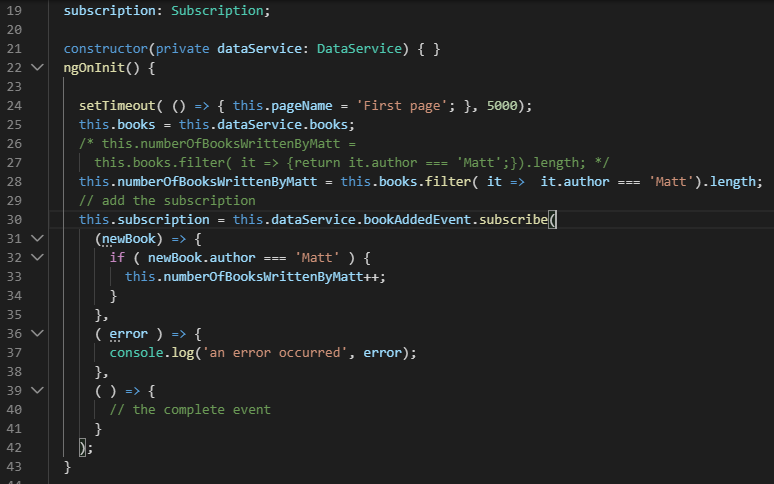
**The 3rd event is the complete event**

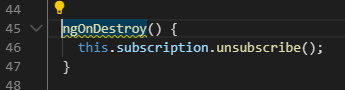


**So, a subscriber has 3 events!**

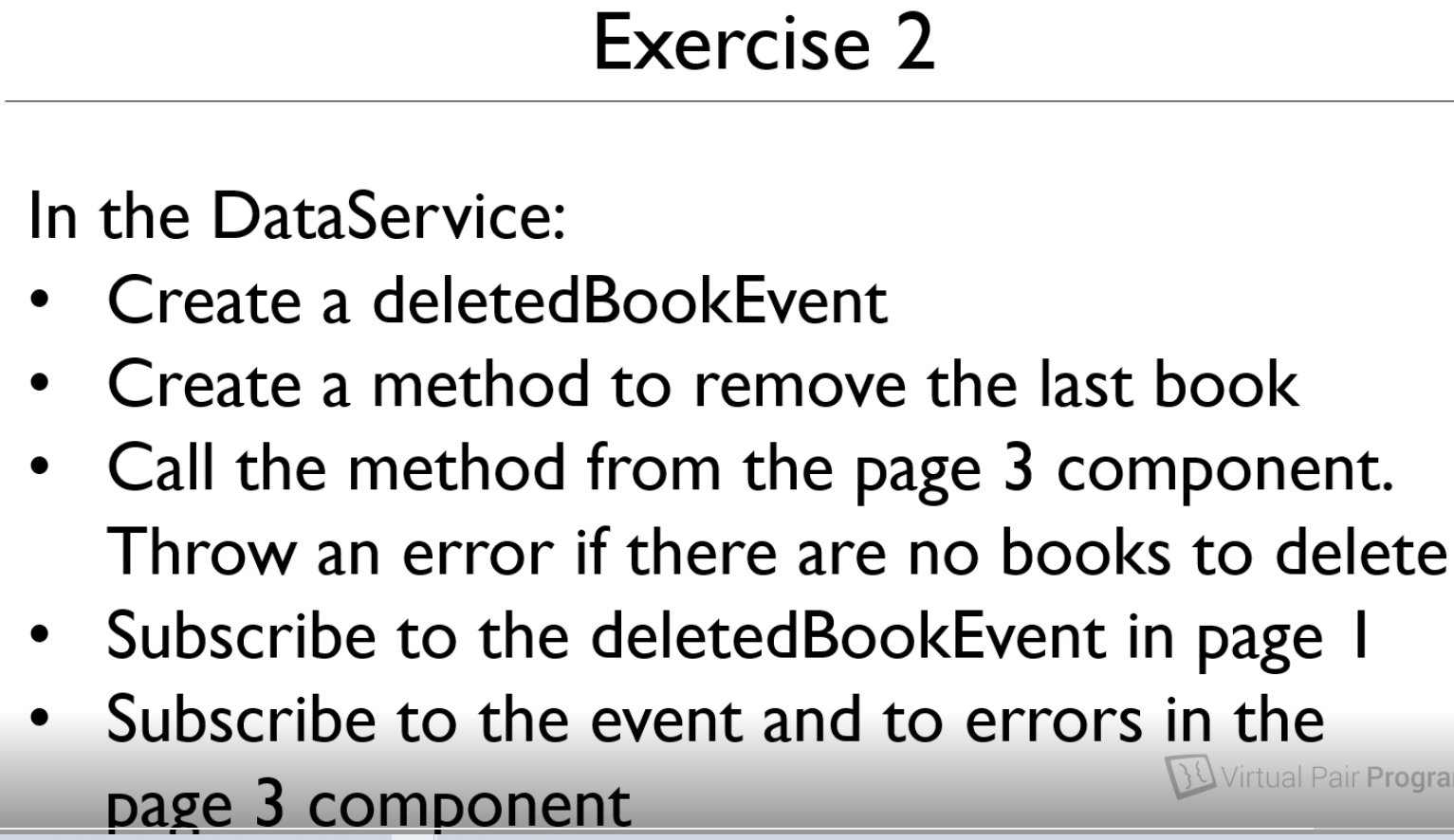
**Unsubscribing an Observer: To prevent memory leak errors This is all done in the page1.ts file**







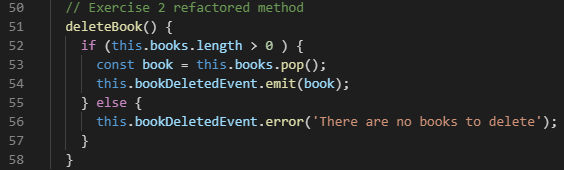
**EXERCISE 2: On the Observer Pattern**



**Add the deletedEvent to DataService.ts**



**DataService refactor**



**Add subscription2 to the page1.ts**





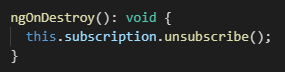
**Add the onDestroy to page1.ts**



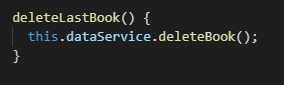
**Add the onDestroy to page3.ts**



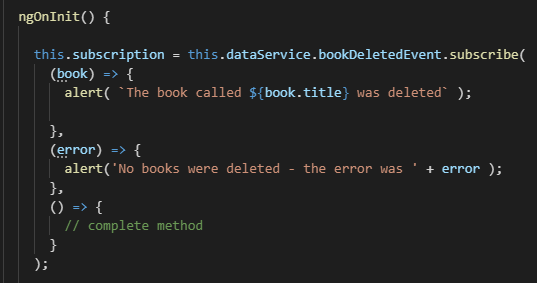
**Add the ngOnDestroy to unsubscribe for page3.ts**



**Refactor the deleteLast() page3.ts**



**Add to page3.ts subscription implementation**



**Tested and it works!**

**Bug: if you delete all, the Matt books does not delete!!**