

Angular Js RxJs- Observables



Agenda

- **Creating First Observable**
- **Subject**
- **Behavior Subject**
- **Replay Subject**

Creating First Observable

- **Import Observable**
 - import { Observable } from 'rxjs/Observable';
 - import { of } from 'rxjs/observable/of';
- **Don't import full RxJS otherwise slow loading**

Creating First Observable

- **Observable name should end with \$- just standard practice**
- **Observable is function and takes in observer(object with 3 methods- next,error and complete)**
- **Observer object listens to button,event,data stream**
- **When we subscribe to observable if subscribe is open even when we leave page**
- **Use ngOnInit() life cycle hook**
- **Use ngOnDestroy()- do this when you leave component- so no memory leaks here**

Creating First Observable

- `observable$;`
-
- `ngOnInit() {`
- `this.observable$ = Observable.create((observer) => {`
- `observer.next(1);`
- `observer.next(2);`
- `observer.next(3);`
- `observer.complete();`
- `});`
- `this.observable$.subscribe(`
- `value => console.log(value),`
- `err => {},`
- `() => console.log('this is the end')`
- `);`
- `}`
-
- `ngOnDestroy() {`
- `this.observable$.unsubscribe();`
- `}`

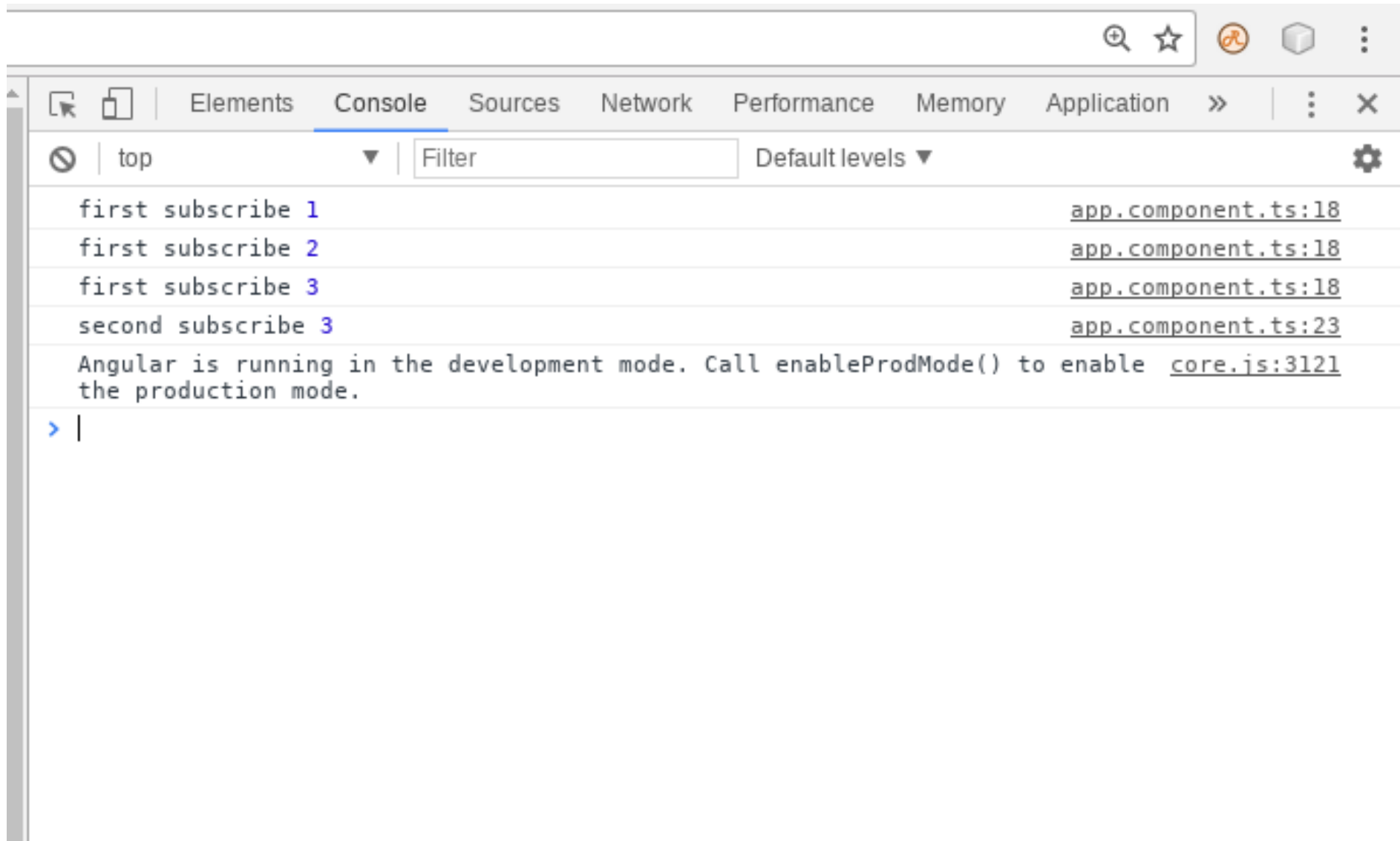
Subject

- Each Observable provides Single Observer instance means no reuse.
- How do we access observer outside like
- `ngOnInit() {`
- `this.observable$ = Observable.create((observer) => {`
- `observer.next(1);`
- `observer.next(2);`
- `observer.next(3);`
- `observer.complete();`
- `});`
- **`observer.next(4);`**
- `this.observable$.subscribe(`
- `value => console.log(value),`
- `err => {},`
- `() => console.log('this is the end')`
- `);`
- `}`

Subject

- **Subject is both an observable and observer**
- **Once we define subject we can access next, err or complete from anywhere.**
- **Subjects are sharable but not reusable. Once completed, not available to subscribe.**
 - `this.mySubject$.complete();`
 - `this.mySubject$.unsubscribe();`
- **Complete silently finishes task, use unsubscribe to get error message if we accidentally use subject again.**

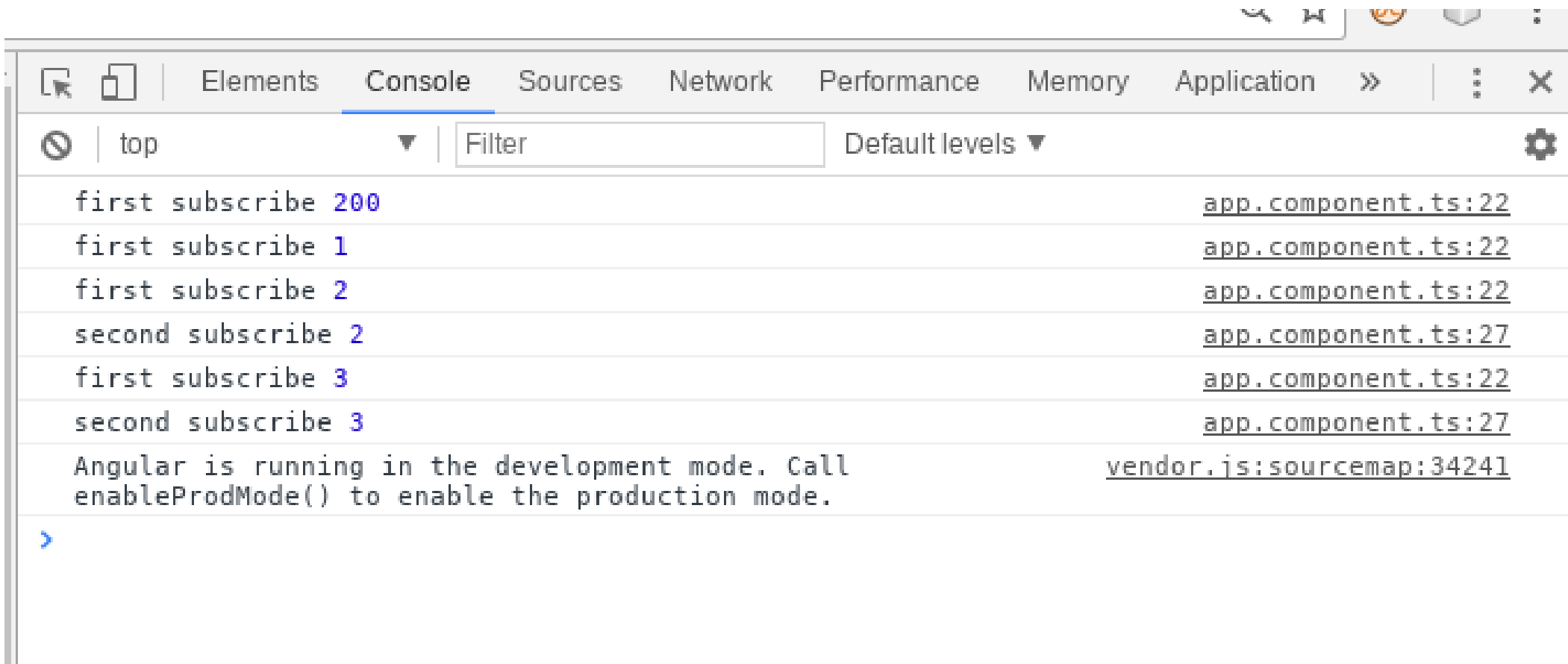
Multiple Subscribe



Behavior Subject

- **How to get the last value passed in?**
- **Behavior Subject are like subject except they need starting value and holds most recent value for new subscribe.**

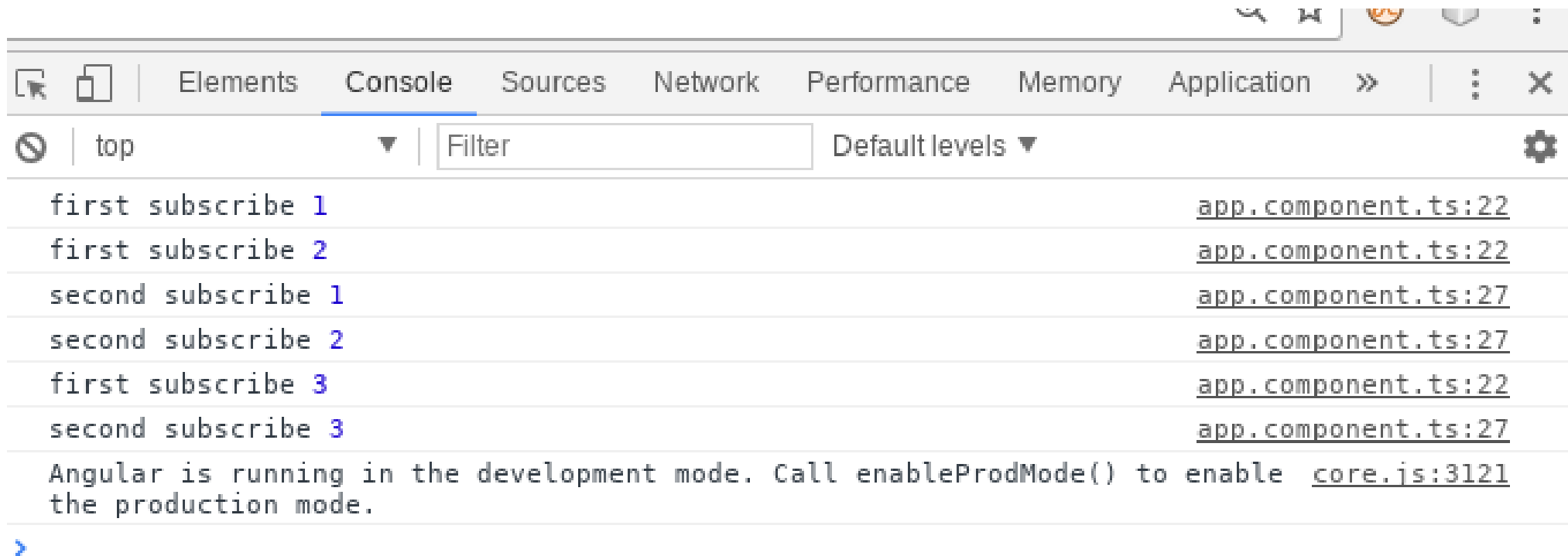
Behavior Subject



Replay Subject

- **How to get all values just passed through- not just the last one?**
- **Replay subject does not require initial value can be used to get all passed values.**

Replay Subject



Resources

- [**https://angular.io/**](https://angular.io/)
- [**https://rxjs-dev.firebaseapp.com/**](https://rxjs-dev.firebaseapp.com/)
- [**http://reactivex.io/**](http://reactivex.io/)

