How to debug Observable values in Angular2 / Typescript?

Asked 2 years, 10 months ago Active 2 months ago Viewed 8k times

3

3

I have followed the tutorial for angular 2 and have a search functionality that renders a list of heroes asynchronously.

```
<div *ngFor="let hero of heroes | async">
    {{hero.name}}
</div>
```

In the component I have observable heroes:

```
heroes: Observable<Hero[]>;
```

Now I have implemented similar functionality in my application by I don't see anything and I don't see any errors either. I opened the debugger in Chrome and tried to check the value of heroes, but it's just some Observable wrapper of course.

Is there any way to see the current/last or some value in the debugger or maybe there is some other technique to debug such issues?

angular typescript rxjs observable rxjs5

edited Jun 27 '17 at 15:46
martin
52.7k 14 111 154

asked Oct 26 '16 at 7:48
Ilya Chernomordik
11.1k 7 52 101

1 Hopefully, some tools to aid debugging RxJS will be built. Until then, you might find this useful: staltz.com/how-to-debug-rxjs-code.html And maybe this answer: stackoverflow.com/a/38597548/6680611 – cartant Oct 26 '16 at 7:53

5 Answers

8

✓

- https://github.com/Reactive-Extensions/RxJS/blob/master/doc/gettingstarted/testing.md#debugging-your-rx-application
- How to debug rxjs5?
- http://staltz.com/how-to-debug-rxjs-code.html
- https://react.rocks/example/rxvision
- http://jaredforsyth.com/2015/03/06/visualizing-reactive-streams-hot-and-cold/

edited May 23 '17 at 12:17
Community ♦
1 1

answered Oct 26 '16 at 8:12
martin
52.7k 14 111 154

So there is no way to use the debugger for that purpose if I understood correctly? – Ilya Chernomordik Oct 26 '16 at 8:26

1 @IlyaChernomordik Of course you can use debugger. The problem is that it's not very helpful because there're so many nested calls and so many anonymous Observables and function that it's very hard keep track of what's going on inside. This is described also in the doc github.com/Reactive-Extensions/RxJS/blob/master/doc/… – martin Oct 26 '16 at 8:29

5

In RxJS v6+, the `tap()` operator has replaced `do()`. So it will look more like this now:

```
someObservable$.pipe(
  map(x => x.whatever),
  tap(whatever => console.log(whatever)),
)
```

answered Feb 6 at 14:02
Harry
51 1 3

2

First of all, if you're using typescript consider:

```
heroes: Observable<Array<Hero>>;
```

```
heroes.subscribe((v) => console.log('got new heroes list: ', v));
```

answered Oct 26 '16 at 8:02

Meir
10.7k 3 24 40

So back to console logging and no debugging options? P.S. Why is Array<Hero> better than Hero[]? This is directly from Angular2 tutorial. – Ilya Chernomordik Oct 26 '16 at 8:09

No pain no gain :-) But even stalz relate to the hardship of debugging. As for Array<Item>, I find it more readable and consistant with the typed approach. Imagine Array<Array<Item>>, Item[][]? (Item[])[]? it gets messy. – Meir Oct 26 '16 at 8:30

1

You can use the **do()** operator EX :

```
this.http
        .get('someUrl', options)
        .map(res => res.json())
        .do(
            data => {
                console.log(data);
            },
            error => {
                console.log(error)
            }
        )
```

If nothing happens inside the do() functions it means you are not subscribed to that observable, remember that observables are waiting for a subscription to start doing something.

answered Oct 26 '16 at 8:28

tibbus
3,690 2 19 32

If you want to do some `console.log` debugging, I wrote a little helper `logValues()` in `s-rxjs-utils`. You can use it like this (from the

```
of(1, 2).pipe(logValues()).subscribe();
// prints using console.log:
// [value] 1
// [value] 2
// [complete]
```

You can also pass it a string to print along with the values, and specify the log level.

answered Jun 15 at 23:14

Eric Simonton
3,307 2 22 40