

Using an array from Observable Object with ngFor and Async Pipe Angular 2

Asked 3 years, 1 month ago Active 2 months ago Viewed 99k times

I am trying to understand how to use Observables in Angular 2. I have this service:

62

```
import {Injectable, EventEmitter, ViewChild} from '@angular/core';
import {Observable} from 'rxjs/Observable';
import {Subject} from 'rxjs/Subject';
import {BehaviorSubject} from 'rxjs/Rx';
import {Availabilities} from '../availabilities-interface'

@Injectable()
export class AppointmentChoiceStore {
  public _appointmentChoices: BehaviorSubject<Availabilities> = new
  BehaviorSubject<Availabilities>({"availabilities": [], "length": 0})

  constructor() {}

  getAppointments() {
    return this.asObservable(this._appointmentChoices)
  }
  asObservable(subject: Subject<any>) {
    return new Observable(fn => subject.subscribe(fn));
  }
}
```

This BehaviorSubject is pushed new values as so from another service:

```
that._appointmentChoiceStore._appointmentChoices.next(parseObject)
```

I subscribe to it in the form of an observable in the component I want to display it in:

```
import {Component, OnInit, AfterViewInit} from '@angular/core'
import {AppointmentChoiceStore} from '../shared/appointment-choice-service'
import {Observable} from 'rxjs/Observable'
import {Subject} from 'rxjs/Subject'
import {BehaviorSubject} from 'rxjs/Rx';
import {Availabilities} from '../shared/availabilities-interface'
```

```

declare const moment: any

@Component({
  selector: 'my-appointment-choice',
  template: require('./appointmentchoice-template.html'),
  styles: [require('./appointmentchoice-style.css')],
  pipes: [CustomPipe]
})

export class AppointmentChoiceComponent implements OnInit, AfterViewInit {
  private _nextFourAppointments: Observable<string[]>

  constructor(private _appointmentChoiceStore: AppointmentChoiceStore) {
    this._appointmentChoiceStore.getAppointments().subscribe(function(value) {
      this._nextFourAppointments = value
    })
  }
}

```

And the attempt to display in the view as so:

```

<li *ngFor="#appointment of _nextFourAppointments.availabilities | async">
  <div class="text-left appointment-flex">{{appointment | date: 'EEE' |
uppercase}}

```

However, availabilities is't yet a property of the observable object so it errors out, even though I define it in the availabilities interface as so:

```

export interface Availabilities {
  "availabilities": string[],
  "length": number
}

```

How can I display an array asynchronously from an observable object with the async pipe and *ngFor? The error message I get is:

```

browser_adapter.js:77 ORIGINAL EXCEPTION: TypeError: Cannot read property
'availabilities' of undefined

```

typescript

 angular

rxjs

observable

edited Oct 12 '17 at 8:12

asked Jun 7 '16 at 3:07



mihai

1,530

2

22

46



C. Kearns

436

2

8

17

What's the actual error message? – [Günter Zöchbauer](#) Jun 7 '16 at 3:11

edited to add error – [C. Kearns](#) Jun 7 '16 at 3:21

with the latest angular-rc1 the syntax is `*ngFor="let appointment of _nextFourAppointments.availabilities | async">` – [Jagannath](#) Jun 7 '16 at 3:39

this is true, but it is not causing the error. it simply throws a warning. – [C. Kearns](#) Jun 7 '16 at 3:40

2 I believe there is a typo somewhere. The error says `availabilities` while there should be `availabilities` – [Ivan Sivak](#) Jun 7 '16 at 3:46

1 Answer



Here's an example

112



```
// in the service
getVehicles(){
    return Observable.interval(2200).map(i=> [{name: 'car 1'},{name: 'car 2'}])
}

// in the controller
vehicles: Observable<Array<any>>
ngOnInit() {
    this.vehicles = this._vehicleService.getVehicles();
}

// in template
<div *ngFor='let vehicle of vehicles | async'>
    {{vehicle.name}}
</div>
```

edited Jun 7 '16 at 8:59

answered Jun 7 '16 at 8:48



Relu Mesaros


2,636

2

15


30

my get function is returning a subject though: `public _appointmentChoices: Subject<any> = new Subject() getAppointments() { return this._appointmentChoices.map(object=>object.availabilities).subscribe() }` , in the controller when I set it equal, I get the error:

browser_adapter.js:77Error: Invalid argument '[object Object]' for pipe 'AsyncPipe' , how do I turn the subject into an observable? – [C. Kearns](#) Jun 7 '16 at 16:27 

```
public _appointmentChoices: Subject<any> = new Subject()
getAppointments() { return
  (this._appointmentChoices.map(object=>object.availabilities).asObservable()) } }

```

this gives me the error: property asObservable does not exist on type observable , but _appointmentChoices is a Subject ? – [C. Kearns](#) Jun 7 '16 at 16:56 

It already was an observable! I just needed to subscribe to it! – [C. Kearns](#) Jun 7 '16 at 16:58

I had an additional problem with integrating subjects. Here's a StackBlitz using observables and subjects: stackblitz.com/edit/subject-as-observable-list-example – [IceWarrior353](#) Jul 23 '18 at 20:12

Got a question that you can't ask on public Stack Overflow? [Learn more](#) about sharing private information with Stack Overflow for Teams.

