Deserialise dates properly in angular 5

Asked 1 year, 2 months ago Active 1 year, 2 months ago Viewed 941 times

2

Angular 5 does not seem to properly deserialise json from the API into dates, at least not in an array. I have a model like this with an array of dates:

```
    export interface ProcessDefinition {
    _id?: string;
    processDefinitionName: string;
    myDates: Date[];
}
```

I have a service class that uses HttpClient to return an observable like so:

```
    public getItems(): Observable<ProcessDefinition[]> {
    let url = this.apiUrl + this.entityName + '/';

    this.loggerService.log(`Getting items with observable via http.`);
    this.loggerService.log(`URL: ` + url);

    return this.httpClient.get<ProcessDefinition[]>(url);
}
```

I call that service from my component, like so:

```
public loadProcessDefinitionData(): void {
    this.processDefinitionService.getItems().subscribe(items => {

        this.processDefinitions = items;

        // Does not help
        // this.processDefinitions.forEach(processDef =>
        //     processDef.day1dates.forEach(day1d =>
        //         day1d = new Date(day1d))
```

```
            // Nope
            this.currentProcessDefinition.day1dates.forEach(dat => dat = new Date(dat));

            // Would like this to work, confirming it's a real date.
    console.log(JSON.stringify(this.currentProcessDefinition.day1dates[0].getMonth()));

        });
    }
```

Above shows my hamfisted attempts to convert to a 'real date' using the 'new Date' approach discussed in other questions. I'm sure it would be straightforward for someone more familiar with the syntax.

What I want is for the ProcessDefinition[] observable to contain a myDates array of real dates, confirmable by calling getMonth() successfully on one. Ideally this conversion would happen in the service so it only has to be in one place.

Thanks

angular typescript type-conversion observable

asked Jul 1 '18 at 9:36
Glinkot
1,547 9 34 60

Instead of using `forEach` , use `.map` , so: `this.currentProcessDefinition.day1dates = this.currentProcessDefinition.day1dates.map(dat => new Date(dat))` – user184994 Jul 1 '18 at 9:42

That does it! Pls put it as an answer and I'll mark it as such. If you know the syntax for adding this to the service class instead of my component, I'd appreciate it. ie do the map on the "return this.httpClient.get<ProcessDefinition[]>(url);" – Glinkot Jul 1 '18 at 9:52

Sure. Which version of Angular / RXJS are you using? – user184994 Jul 1 '18 at 9:53

Angular5, Rxjs 5.5.6, but I could upgrade if needed. I had a prior http service using the Http provider from angular/http, but recently changed it to the HttpClient version I show above. The original code was: public getItems(): Observable<ProcessDefinition[]> { let url = this.apiUrl + this.entityName + '/'; return this.http.get(url) .map(response => response.json() as ProcessDefinition[]) .do((items) => { if (this.logging) { this.loggerService.log(`Got ${items.length} items.`) }; }); } – Glinkot Jul 1 '18 at 10:02

Okay. The answer I've just posted was done using rxjs6, but should be compatible. Let me know if there are any issues. – user184994 Jul 1 '18 at 10:05

1 Answer

3

Instead of using `forEach` , use the arrays [`.map`] function, so:

```
this.currentProcessDefinition.day1dates =
this.currentProcessDefinition.day1dates.map(dat => new Date(dat))
```

Alternately, if you want to do it in your service, you can do it like so:

```
import { map } from 'rxjs/operators'

getItems() {
  let url = this.apiUrl + this.entityName + '/';

  return this.http.get(url).pipe(
    // Map through each item in res, and format the object
    map((res) => res.map(item => this.formatDates(item)))
  )
}

formatDates(results) {
  // Map through each date and replace with Date objects
  results.day1dates = results.day1dates.map(dat => new Date(dat));
  return results;
}
```

In this code, we're piping the Observable through it's own `.map` function, to transform the results. In your components, you can then subscribe as usual.

edited Jul 1 '18 at 10:11

answered Jul 1 '18 at 10:04
user184994
12.5k 1 21 30

I get 'property 'map' does not exist on type 'Response'. I've changed my import from the old 'import 'rxjs/add/operator/map'; to your version, and tried it with http as an instance of Http and also HttpClient, same result. Any tips? – Glinkot Jul 1 '18 at 10:18

Change the `http.get` to match your original, i.e. `http.get<ProcessDefinition[]>` and it should work. The response needs to be typed as an array