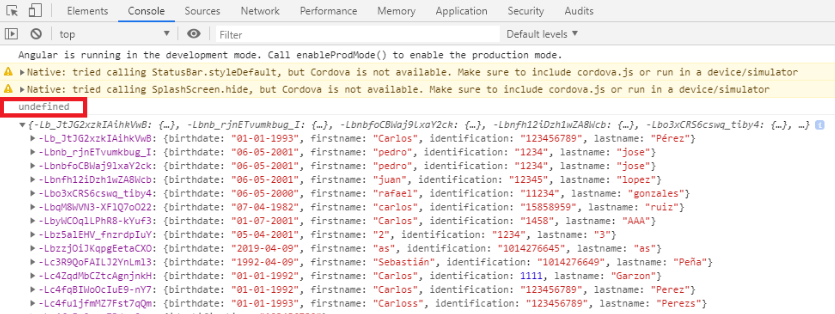
This repost explains why I didn’t manage to Access the information returned by the GET request after trying multiple methods including the official Angular documentation, an official Angular Http tutorial and other methods found in different forums.

All the different ways of using the GET request I tried returned an Observable that had to be converted to a usable Object in order to be usable. But with every method used to try to convert the result would be “Undefined” as show in the image below where “undefined” is the attempt to print the data after subscribing to the Observable and the information below is the observable being printed.



The code snippets below show the different methods used.

GET using a pipe and map from the rsjx library, this particular method chow the object and then it is subscribed to from where the method is called to show only the information without any other headers so it’s easier to use.

this.http.get('https://testbankapi.firebaseio.com/clients.json')

.pipe(

map((res: any) => {

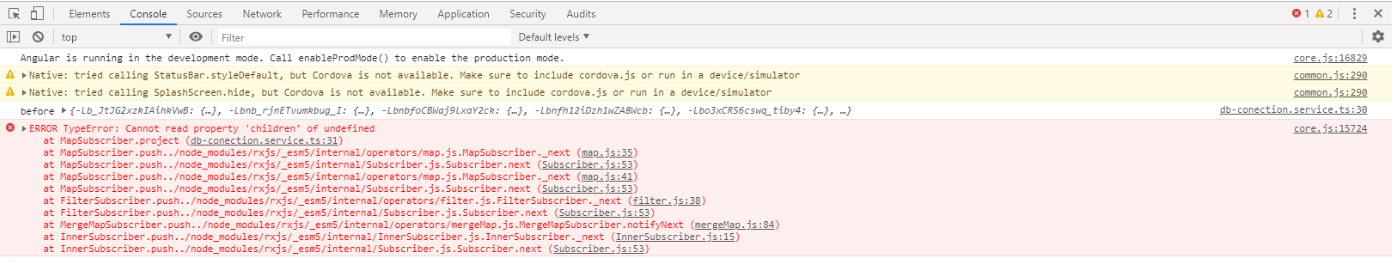
console.log('before', res);

return res.data.children;

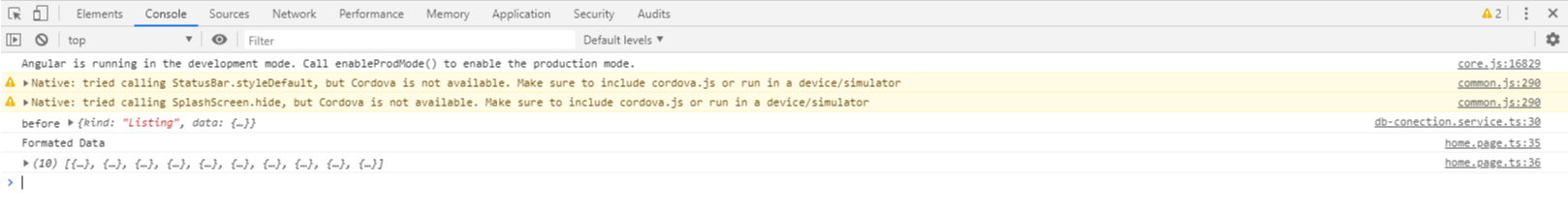
})

);

The following image shows the result where the before print is the information returned by the GET request and the error is from where I tried to access to it.



I tried using a different end point with the same method and the result was that I manage to subscribe to the response and print it.



Another method I tried was to create an interface with the same kind of value fields as the json has and named it “Person”, then created “customers”, which is an array of Person type.

export interface Person{

birthdate : string,

firstname : string,

identification : string,

lastname : string

}

customers : Person[]=[];

Then made the GET request and tried to save the response from the request to the customers array but the result was an empty array.

this.http.get("https://testbankapi.firebaseio.com/clients.json").subscribe((res)=>{

console.log(res);

this.customers = res;

}, error => {

console.log('Error', error);

});

The official methods from the Angular documentation ended up giving the “Undefined” result. The methods that I tried were:

* Getting JSON data (from: <https://angular.io/guide/http>).
* Standalone pipe function (from: <https://angular.io/guide/rx-library>).
* Http Angular tutorial (from: <https://angular.io/tutorial/toh-pt6>).

It’s also worth mentioning that at some point during the development, the GET request got blocked by CORS (Cross-origin resource sharing) and when I found a workaround it the response got blocked by CORB (Cross-origin Read Blocking).