

How do I use observable and observer in angular?

▲ I come from a synchronous programming background and I am having a hard time understanding observables.

1 Here is an extract of my service/provider (Ionic 2 project)

▼

```
return this.http.get(`${this.serverApi}`)  
  .map(res => <Response[]>res.json());
```



1 and I will subscribe to that from the LoginPage . I have several questions regarding this.

1. Does the above code return an observable/ observer even if I didn't declare it as such?
2. The response is JSON. How do I do some checking/processing of the JSON and perform some action like if

```
res.auth_token==true
```

then do

```
localStorage.setItem(res.auth_token)
```

I believe it should be done in the provider class. Just a typical hint/example will be awesome.

3. Does the request actually happen when it hits the subscribe method?
4. [Creating and returning Observable from Angular 2 Service](#) mentions Subject and ReplaySubject. Should I use them instead?

javascript



angular

rxjs

observable

edited May 23 '17 at 12:02



Community ♦

asked Jan 9 '17 at 12:36



TradeRaider

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

1 Answer



3



1. The code will return an `Observable`
2. You can change the callback body to a block and add as much code as you want

```
return this.http.get(`${this.serverApi}`)  
  .map(res => {  
    let x = <Response[]>res.json();  
    // do something with x  
    res.auth_token == true;  
    return res; // with a block body an explicit `return` is required  
  });
```

3. Yes, calling `.subscribe()` will execute the `http.get()` and then all subsequent operators when the response arrives. Without `.subscribe()` nothing will happen.

Note: Some Angular APIs expect an `Observable`, in this case you must not call `subscribe()` because `subscribe()` returns a `Subscription`, not an `Observable`.

4. it depends what you try to accomplish. If you use `http.get()`, you already get an observable and there is no need for `Subject`, `ReplaySubject`, or any of the other possibilities.

answered Jan 9 '17 at 12:45



[Günter Zöchbauer](#)

353k 81 1109 1017

Thank you. This makes a lot of sense. Regarding `ReplaySubject`, when should I use it? My app makes http get, post, put and delete requests btw. – [TradeRaider](#) Jan 9 '17 at 12:59

`ReplaySubject` is useful if subscribers that subscribe later should also receive all previously emitted events. I don't know of a good use case just now,

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 