[Observable](https://www.udemy.com/the-complete-guide-to-angular-2/learn/lecture/6656534" \l "overview" \t "_blank) (Rx-Js)

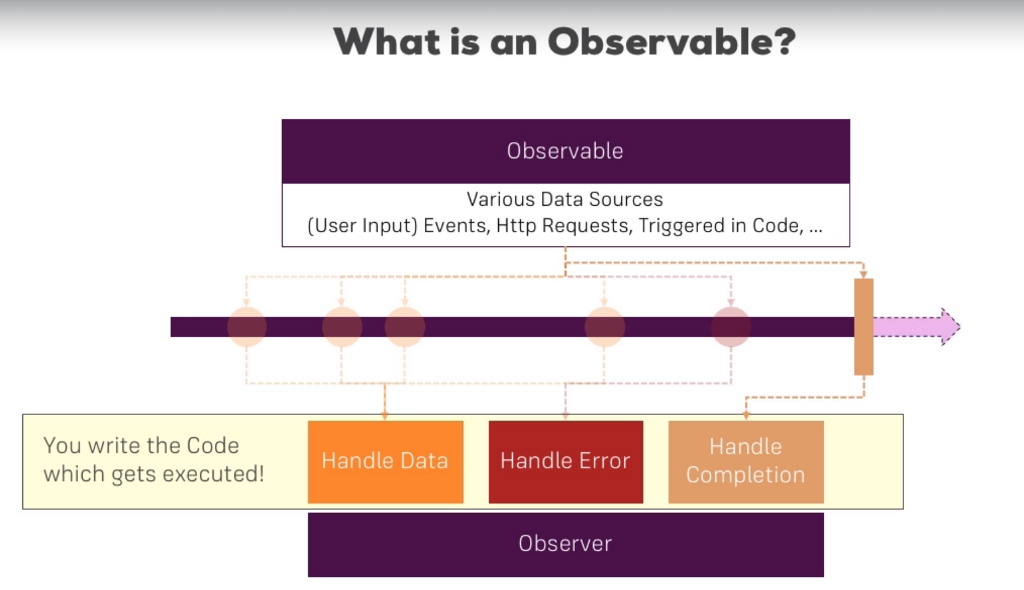
What is?

* Basically a DataSource, Strem of data

Observable --------------------------> Observer

On subscribe, you handle packages in three way:

|  |  |
| --- | --- |
| -Handle Data | (Singolo evento) |
| -Handle Error | (In caso di errore) |
| -Handle Completion | (Al termine) |



Ex:

This.route.params.subscribe((evt)=>{…});

Create new Observable:

RxJs is mandatory. Check package.json

Remember to unsubscribe observable on destroy component.

See Interval example.

Operators:

To concat more operators, use pipe() method.

Ex:

Observable.pipe(  map (data => { return data; }),

operator2(){},

operator3(){}…).subscribbe(…);

[Elenco operatori](https://www.learnrxjs.io/operators/)

Esempio per evitare che molteplici input dell'utente facciano troppe chiamate:

import { fromEvent } from 'rxjs';

import { debounceTime, map } from 'rxjs/operators';

// elem ref

const searchBox = document.getElementById('search');

// streams

const keyup$ = fromEvent(searchBox, 'keyup')

// wait .5s between keyups to emit current value

keyup$.pipe(

map((i: any) => i.currentTarget.value),

debounceTime(2000)

)

.subscribe(console.log);

Stream line

(Qui vengono emessi gli eventi)

Observable

Observer