Rxjs:

RxJs Operators:

There are multiple operators available to transform the data returned from the observables.

Types of Operators:



The ‘Pipe’ operator is used to create the pipeline for transformations using multiple operators. The output of one operator becomes input for next operator.

Let MyObservable = of([1,2,3,4,5,6]); //The of() will produce an observable

MyObservable.Pipe(

mergeMap(resp => resp.response),

filter(obj => obj.ShippedDate > ‘2019-12-19’),

catchError(err => throw ‘something went wrong - ${err.message}’)

)

.Subscribe(val => Console.Log(val);

Note that we created a pipeline having 2 operators mergeMap (flatten the data from api response) and filter (filters the values against a condition).

Subjects:

Subjects are both observables and observers. Like Observables they can produce values and pass those values to observers via it’s subscribe method. Also like observers they have next(), error() and complete() methods to handle the values received by observables. This way they act as both observers and observables.

The imp. Char. Of Subjects is that they maintain a list of Observers (being a Observable). It can push values to all of its subscribers. This is called multicast subjects.

A screenshot of a cell phone

Description automatically generated

Let’s see an example where a subject maintains a list or array of observers and calls them when it receives the values from a source observable.

A screenshot of a computer screen

Description automatically generated

A screenshot of a cell phone

Description automatically generated