**Observables:**

**Reference URLs:**

[Angular 7 | Observables - GeeksforGeeks](https://www.geeksforgeeks.org/angular-7-observables/)

[Angular Promises Versus Observables | Syncfusion Blogs](https://www.syncfusion.com/blogs/post/angular-promises-versus-observables.aspx#:~:text=Promises%20deal%20with%20one%20asynchronous,over%20a%20period%20of%20time.&text=Emit%20multiple%20values%20over%20a,single%20value%20at%20a%20time.)

<https://www.javatpoint.com/observable-in-angular>

<https://appdividend.com/2022/02/03/angular-observables/>

<https://www.c-sharpcorner.com/article/subject-and-behavior-subject-in-angular-8/>

<https://dev.to/this-is-learning/rxjs-creation-operators-6bh>

<https://dev.to/this-is-learning/rxjs-filtering-operators-pe1>

<https://rxjs.dev/guide/operators>

<https://codecraft.tv/courses/angular/http/http-with-observables/>

<https://codecraft.tv/courses/angular/reactive-programming-with-rxjs/observables-and-rxjs/>

<https://blog.logrocket.com/using-rxjs-operators-consume-observables/>

[RxJS Observable Pipe, Subscribe, Map and Filter Examples with Angular 9/8 | Techiediaries](https://www.techiediaries.com/observable-pipe-subscribe-map-and-filter-examples/#:~:text=RxJS%27%20pipe()%20is%20both,and%20returns%20an%20RxJS%20Observable.)

Observables provide support for data sharing between publishers and subscribers in an angular application. It is referred to as a better technique for event handling, asynchronous programming, and handling multiple values as compared to techniques like promises.  
A special feature of Observables is that it can only be accessed by a consumer who subscribes to it i.e. A function for publishing values is defined, but it is not executed till it is subscribed by the consumer (it can be any component). Consumer can receive notifications till the function runs or till they subscribed.

**Observers:** To handle receiving observable messages, we need an observable interface which consists of callback methods with respect to the messages by observables. Some of the basic methods are as follows: 

* **next**: It is a handler for each message by observable, it may be called zero or more times after execution starts.
* **error:** It is a handler for each error message. An error stops execution of the observable instance.
* **complete:** It is a handler in which the completion of observable execution is notified.

**difference between observable and promise**(observable vs promise)

|  |  |
| --- | --- |
| **Observables** | **Promises** |
| Emit multiple values over a period of time. | Emit a single value at a time. |
| Are lazy: they’re not executed until we subscribe to them using the subscribe() method. | Are not lazy: execute immediately after creation. |
| Have subscriptions that are cancellable using the unsubscribe() method, which stops the listener from receiving further values. | Are not cancellable. |
| Provide the map for forEach, filter, reduce, retry, and retryWhen operators. | Don’t provide any operations. |

RXJS:

**RxJS (Reactive Extensions for JavaScript) is a library for reactive programming using observables that makes it easier to compose asynchronous or callback-based code**.

Subject and BehaviorSubject:

What is a Subject?

Subject is a special type of Observable in RxJs Library in which we can send our data to other components or services. A Subject is like an Observable but can multicast to many observers which means subject is at the same time an Observable and an Observer.

Here are some important points of subjects

* A Subject is a Special type of Observable that allows value to be multicasted to many Observers.
* Subject are like event emitters.
* No Initial Value

1. subject =**new** Subject<datatype>();

What is a Behavior Subject?

Behavior Subject is similar to subject but only difference is that we can set the initial value .

A variant of subject that requires initial value.

1. **const** subject =**new** BehaviorSubject(0); //0 is the initial value.

Difference between Subject and Behavior Subject?

In Subject we cannot set an initial value but in Behavior subject we have to set an initial value otherwise it will throw an error.

**URL for Demo:** [Subject And Behavior Subject In Angular 8 (c-sharpcorner.com)](https://www.c-sharpcorner.com/article/subject-and-behavior-subject-in-angular-8/)

**RxJS Operators:**

**Creation Operators:** Used to create a new Observables.

* [**of**](https://rxjs.dev/api/index/function/of)

The Of operator is used to convert an argument to an observable.

For more details, please refer the following URL:

[RxJS - Creation Operators - DEV Community](https://dev.to/this-is-learning/rxjs-creation-operators-6bh)

**Filtering Operators:**

These operators are used to filter our sources.

* [**filter**](https://rxjs.dev/api/operators/filter)

Filter items emitted by the source Observable by only emitting those that satisfy a specified predicate (condition).

For more details, please refer the following URL:

[RxJS - Filtering Operators - DEV Community](https://dev.to/this-is-learning/rxjs-filtering-operators-pe1)