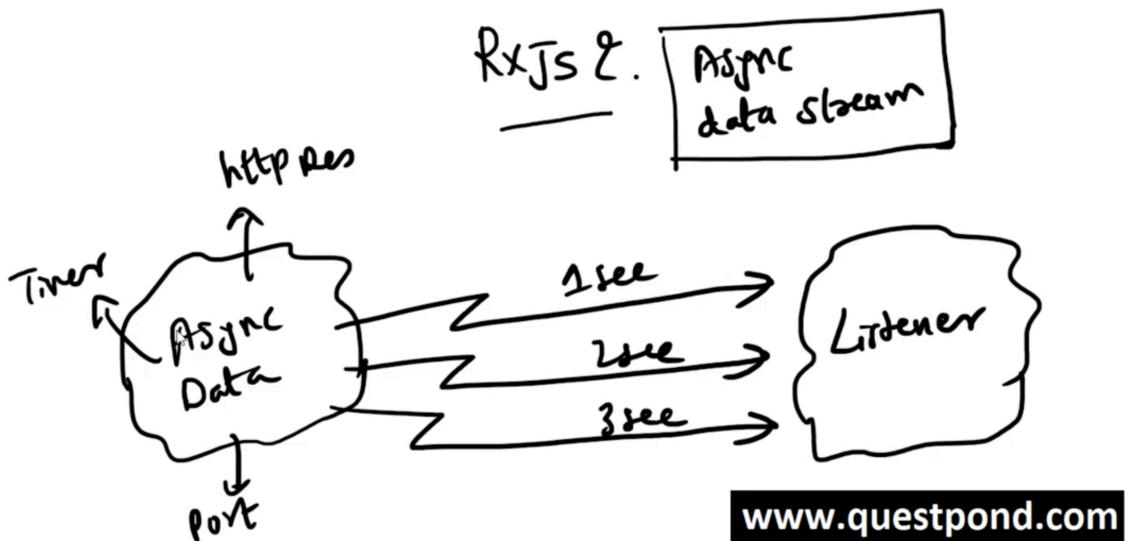


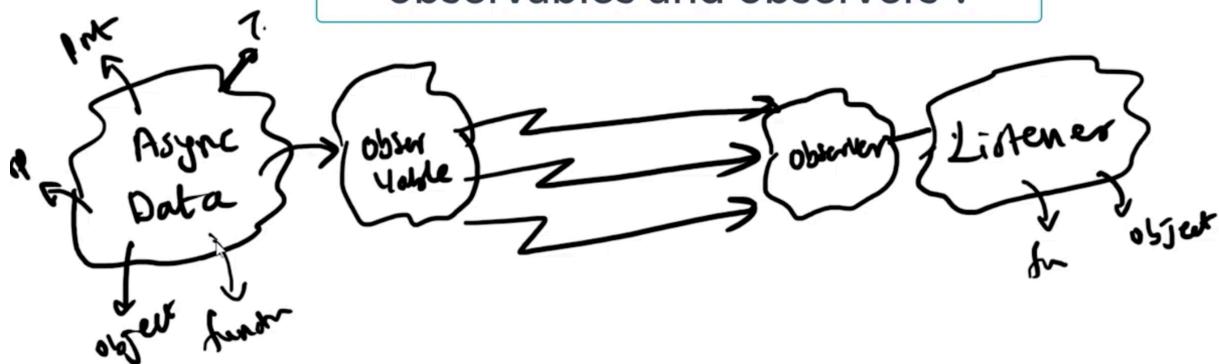
RxJS interview questions:

1. What is the full form of RxJS?
 - Reactive extensions for javascript
2. What is the purpose of RxJs?
 - The only big purpose of RxJs is to handle asynchronous data stream easily in javascript.



3. What are observers and observables?

Question no: 3:- What are observables and observers ?



- Observable represents an asynchronous data streams and observer represent listener who is actually interested to hear this data stream.
- Observable and observers are nothing but they are RxJs objects.
- Observable represents an asynchronous data streams and observer is nothing but a function which is subscribe to that observable and listens to that data stream and receive the data stream.

4. Explain the use of subscribe method in RxJs with sample code?

```

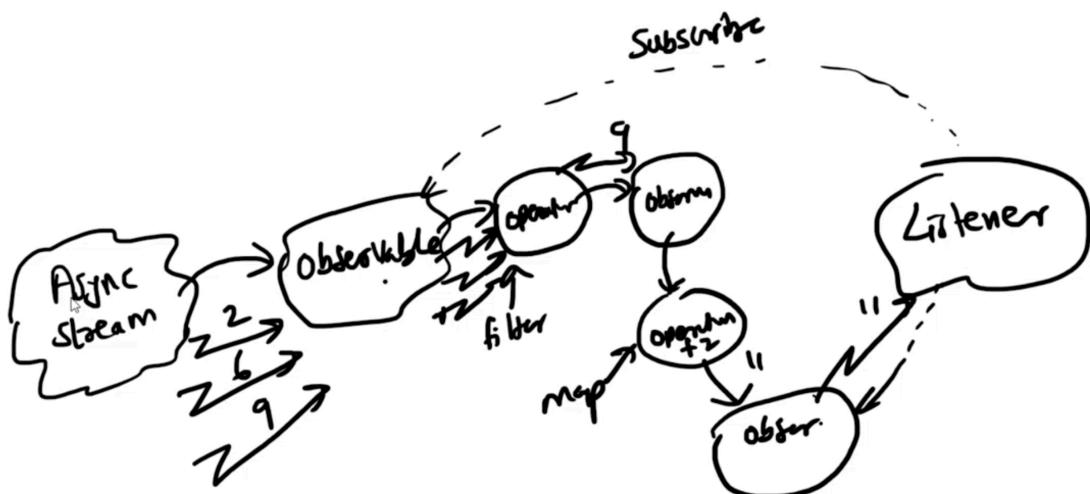
1 import {Observable} from 'rxjs';
2 > function AsyncStream(observer){ ... }
6 }
7
8 var observ = Observable.create(AsyncStream);
9
10 observ.subscribe(res=>Listener(res));
11
12 function Listener(res){
13   |   console.log(res);

```

- Step 1 first create an observable from that stream
 - step 2 you subscribe to that stream and you attach your listener which can be a function which can be an object
5. How to unsubscribe in RxJs?
- first thing is get hold of the subscription object - var subscription = pbserv.subscribe(res=>Listener(res));
 - and then say that hold object name.unsubscribe -
- ```

function Listener (res){
 console.log(res);
 subscription.unsubscribe();
}

```
6. Explain concept of operators with sample code?



- Operators are nothing but they are small piece of logic which actually converts an observable streams to another observable streams
- If you want to go and plug in preprocessing logic before the data comes to the listener I will use operator.

```
var mainstreamobservable = Observable.create(AsyncStream);
var step1stream = mainstreamobservable.pipe(map(x=>Math.round(x)));
var step2stream = step1stream.pipe(filter(x=>x>4));
-
```

7. How to install RxJs?

- npm i rxjs

8. Differentiate between RxJs and Promises?

- In RxJs, observables return stream of data, on other side promise return single value.
- In RxJs stream you can subscribe and later if you are not interested you can stop the stream and do unsubscribe, but in promise you can not cancel the promise.

9. In angular where you used RxJs?

- Maximum time I have seen in project RxJs is used with HTTP calls, because the response of the HTTP calls is in the form of stream.
- If you see to the httpClient which belongs common http module, when it make a POST call or GET call it returns an observable, and to this observable we are calling the subscribe we are saying the subscribe to the stream

10. What kind of operators have you used still now?

- Map :- Transforms data in a observable in to a different format.
- Filter :- Allows data which meets conditions.
- Merge:- This operator will combine multiple Observables into one. So if one of the observables emit a value the combined one will emit as well.
- Concat :- only when observable completes, it will start with the next observable.
- From :- This operator will turn array, promise or iterable into an observable. debounceTime :- discard emitted values if a certain time didn't pass between the last input
- distinctUntilChanged :- only emits a value if it is different than the last one. pluck :- select a property to emit. delay :- emits a value with a delay.

11. why is RxJs called push/reactive not pull/imperative?

- Imperative programming means listener code is responsible to pull stream of data.
- Reactive programming means you register a callback and the stream is responsible to push data.
- Some developers also visualize it as publisher and subscriber model as well.

