**Lesson14 Observable in Angular 4**

**HTTP & Observable**

**1-HTTP is used to retrieve or send data to server via Web API**

Response with using (Observable) instead of use

(Standard Way) Response with Promise

Request HTTP

Server

PC

**2-Observable is part of React.js**

**3-React.js is an API asynchronous programming with observable streams**

**4-Observable is an asynchronous stream of values / Data that we subscribe**

**New Values if the updates happens instantly**

**(To guarantee receive new value all the time even if the updates happen after the server send values to the client)**

**5-Observable works like Array**

**6-Observable update the data dynamically even sending data to the client**

|  |  |
| --- | --- |
| **Observable** | **Promise** |
| **Multiple values** | **Single value** |
| **Cancellable** | **Not Cancellable** |
| **Map , filter , reduce** |  |

**7-as we see that the Observer tell the Observable that there are**

Observable

**some data that the person care about it to update it**

**8-you can update the three optional methods**

**Invoke**

**Subscription**

**Next() : will called when we receive new value**

**Error() :will called when we faced error message**

Observer

**Complete(): will called when the operation is complete**

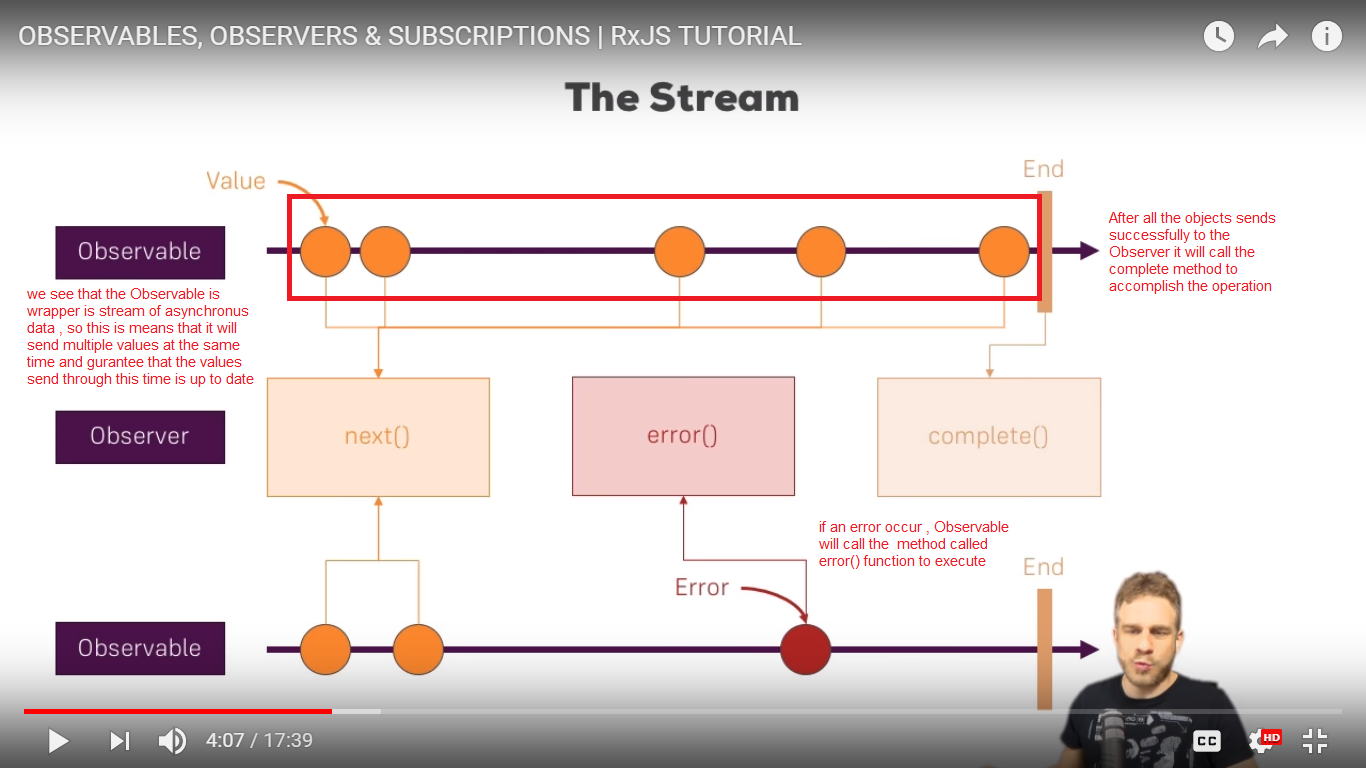
Next()

Error()

complete

**9-we see that the Observer is the only who control the using of the three methods Next (), Error (), complete ()**

**So the Observable will 100% know that the Observer is the right one who have the right to take new value , ….**

****

**1-So the Observable is used to call the next () method on each time require new object, and the Observer take the new object and make submission to the Observable**

**2-if the all Object are sent to the Observer, the Observable execute Complete method which the Observer execute it to close the connection**

**3-if an error occur during the loading data, the Observer throw the error to the error method which the observer handling the error**

**Observable Description Part 2**

**1-HTTP Mechanism**

Server

Browser

EmpList

Database

**Request**

HTTP

**Get**

Emp Service

Emp Detail

**Response**

**Observable**

**2-Observable Mechanism**

**\*we see that the Observable Mechanism**

**1-when the Newspaper send request to the database, the database send response in Observable format, send multiple objects / values**

**2-The Service Newspaper make mapping the Jason object’s to the objects of type Newspaper class**

**3-The Component that Subscribe , it will receive the result only**

Source

Observable

Get

Newspaper Company

Service

**Convert**

**Subscribe**

House

1

House

N

House

2

**Observable: - is sequence of items that arrive asynchronously over time**

**HTTP Call – Single Call**

**Single Item – HTTP responce**

**HTTP, Observable and RXJS**

**1-HTTP Get request from EmpService**

**2-Receive the observable and cast it into the Employee Array**

**3-Subscribe to the observable from the EmpList and EmpDetail**

**4-Assign the Employee Array to local variable**

**RXJS :- is library that enable the Observable**

**1-Reactive Extension for JavaScript**

**2-External Library to work with Observable**

**Steps**

**1-on CMD go to the node\_module under the folder project**

**Cd AngPro/my-app/node\_modules**

**2-install the rxjs under the path**

**npm install rxjs-compat --save**

**3-in the app.module.ts write the following code**

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import {EmpServiceService} from '../app/services/emp-service.service';

import { AppComponent } from './app.component';

import { EmpCompComponent } from './components/emp-comp/emp-comp.component';

//we see that we added the HttpClientModule in the app.module.ts

import {HttpClientModule} from '@angular/common/http';

@NgModule({

declarations: [

AppComponent,

EmpCompComponent

],

imports: [

BrowserModule,

//and we see that we added the HttpClientModule in the imports section

HttpClientModule

],

providers: [EmpServiceService],

bootstrap: [AppComponent]

})

export class AppModule { }

**4-In the emp-service.service.ts write the following code**

import { Injectable } from '@angular/core';

import {Employee} from '../Buisness/Employee';

//we put the Http in the emp-service.service.ts file as the below code

import {HttpClient} from '@angular/common/http';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

@Injectable({

providedIn: 'root'

})

export class EmpServiceService {

constructor(private \_http:HttpClient)

{

}

GetPersons() : Observable<Employee[]>{

//if you want to load external json file you can use it as

let weburl = 'https://jsonplaceholder.typicode.com/users';

//if you want to load internal json file , you must put it

//outside the app folder like the assests

let localurl = 'http://localhost:4200/assets/EmployeeData.json';

return this.\_http.get<Employee[]>(localurl);

}

}

**5-We will create class Employee**

export class Employee

{

Id:number;

name:string;

username:string;

email:string;

phone:string;

website:string;

}

**6-In the emp-comp.component.html we apply the list of object in the html page**

import { Component, OnInit } from '@angular/core';

import {EmpServiceService} from '../../services/emp-service.service';

import {Employee} from '../../Buisness/Employee';

@Component({

selector: 'app-emp-comp',

templateUrl: './emp-comp.component.html',

styleUrls: ['./emp-comp.component.css']

})

export class EmpCompComponent implements OnInit {

//we see that we will create object of type Employee[]

Employees:Employee[];

constructor(public empService:EmpServiceService)

{

this.empService.GetPersons().subscribe(Data => {

this.Employees = Data;

});}

ngOnInit() {}}

**6-in the index.html write the following code**

<ul \*ngFor="let item of users" class="list-group">

<li class="list-group-item">

id : {{item.id}}

</li>

<li class="list-group-item">

{{item.name}} ----> {{item.username}}

</li>

<li class="list-group-item">

{{item.email}}

</li>

<hr>

</ul>

**7-in the Employee.json write the following code**

[

{

"id": 1,

"name": "Leanne Graham",

"username": "Bret",

"email": "Sincere@april.biz",

"phone": "1-770-736-8031 x56442",

"website": "hildegard.org"

},

{

"id": 2,

"name": "Ervin Howell",

"username": "Antonette",

"email": "Shanna@melissa.tv",

"phone": "010-692-6593 x09125",

"website": "anastasia.net"

},

{

"id": 3,

"name": "Clementine Bauch",

"username": "Samantha",

"email": "Nathan@yesenia.net",

"phone": "1-463-123-4447",

"website": "ramiro.info"

},

{

"id": 4,

"name": "Patricia Lebsack",

"username": "Karianne",

"email": "Julianne.OConner@kory.org",

"phone": "493-170-9623 x156",

"website": "kale.biz"

},

{

"id": 5,

"name": "Chelsey Dietrich",

"username": "Kamren",

"email": "Lucio\_Hettinger@annie.ca",

"phone": "(254)954-1289",

"website": "demarco.info"

},

{

"id": 6,

"name": "Mrs. Dennis Schulist",

"username": "Leopoldo\_Corkery",

"email": "Karley\_Dach@jasper.info",

"phone": "1-477-935-8478 x6430",

"website": "ola.org"

},

{

"id": 7,

"name": "Kurtis Weissnat",

"username": "Elwyn.Skiles",

"email": "Telly.Hoeger@billy.biz",

"phone": "210.067.6132",

"website": "elvis.io"

}

]

**Observable with Error Message**

**\*To achieve this following the steps**

**1-In emp-service.service.ts write the following code**

import { Injectable } from '@angular/core';

import {Employee} from '../Buisness/Employee';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

//we put the HttpErrorResponse , catch , throw

import {HttpClient,HttpErrorResponse} from '@angular/common/http';

import 'rxjs/add/operator/catch';

import 'rxjs/add/observable/throw';

@Injectable({providedIn: 'root'})

export class EmpServiceService {

constructor(private \_http:HttpClient){}

GetPersons() : Observable<Employee[]>{

//if you want to load external json file you can use it as

let weburl = 'https://jsonplaceholder.typicode.com/users';

//if you want to load internal json file , you must put it

//outside the app folder like the assests

let localurl = 'http://localhost:4200/assets/EmployeeData1.json';

//we use the catch expression with specific method

return this.\_http.get<Employee[]>(localurl).catch(this.errorHandler);}

errorHandler(error:HttpErrorResponse){

return Observable.throw(error.message || "Server Error");}}

**2-In the emp-component.component.ts**

import { Component, OnInit } from '@angular/core';

import {EmpServiceService} from '../../services/emp-service.service';

import {Employee} from '../../Buisness/Employee';

@Component({

selector: 'app-emp-comp',

templateUrl: './emp-comp.component.html',

styleUrls: ['./emp-comp.component.css']

})

export class EmpCompComponent implements OnInit {

errorMsg:string;

Employees:Employee[];

constructor(public empService:EmpServiceService){

//we put the error message like this

this.empService.GetPersons().subscribe(Data =>

this.Employees = Data,error => this.errorMsg = error);}

ngOnInit() {

}}

**3-in the emp-component.html**

<h3>The Employee's works under you</h3>

<p style="color: red;">{{errorMsg}}</p>

<ul \*ngFor="let item of Employees" class="list-group">

<li class="list-group-item">

id : {{item.id}}

</li>

<li class="list-group-item">

{{item.name}} ----> {{item.username}}

</li>

<li class="list-group-item">

His Email {{item.email}}

</li>

<hr>

</ul>