**Lesson15 Observable in Angular 4 with using Http**

**Notes:-**

**To Apply using the Observable over the using of Http, we do the following:-**

**1-on the app.module.ts we write the following**

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

//we will put the NgModule on the app\_module

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

//we will put the FormsModule on the app\_module

import { FormsModule} from '@angular/forms';

//we will import the Http on the app.module

import { HttpModule } from '@angular/http';

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

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

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

import { EmployeeListComponent } from './Components/employee-list/employee-list.component';

import {EmployeeServiceService} from './Services/employee-service.service';

import { EmployeesComponent } from './Components/employees/employees.component';

import { EmployeeComponent } from './Components/employee/employee.component';

import { ToastrModule } from 'ngx-toastr';

@NgModule({

declarations: [

AppComponent,

EmployeeListComponent,

EmployeesComponent,

EmployeeComponent

],

imports: [

BrowserModule,

BrowserAnimationsModule,

HttpModule,

FormsModule,

ToastrModule.forRoot()

],

providers: [EmployeeServiceService],

bootstrap: [AppComponent]

})

export class AppModule { }

**2-In the EmployeeServie.ts we write the following code:-**

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

import {EmployeeModule} from '../Services/employee-module';

import { Observable } from 'rxjs/Observable';

//we import the Http on the Employee Service

import { Http, Headers, RequestOptions, RequestMethod } from '@angular/http';

//we import the explicit mapping

import 'rxjs/add/operator/map';

@Injectable({

providedIn: 'root'})

export class EmployeeServiceService {

constructor(private \_http:Http) { }

weburl:string = 'http://localhost/WebAPI/api/Employee/';

Employees:EmployeeModule[];

selectedEmployee:EmployeeModule;

errorMsg:string;

GetPersons(){

var headerOptions = new Headers({'accept':'application/json'});

var requestOptions = new RequestOptions({method : RequestMethod.Get,headers : headerOptions});

return this.\_http.get(this.weburl,requestOptions).map(res => res.json());}

GetPerson(id:number){

var headerOptions = new Headers({'accept':'application/json'});

var requestOptions = new RequestOptions({method : RequestMethod.Get,headers : headerOptions});

return this.\_http.get(this.weburl+id,requestOptions).map(res => res.json());}

postEmployee(emp : EmployeeModule){

var body = JSON.stringify(emp);

var headerOptions = new Headers({'Content-Type':'application/json'});

var requestOptions = new RequestOptions({method : RequestMethod.Post,headers : headerOptions});

return this.\_http.post(this.weburl,body,requestOptions).map(x => x.json());}

putEmployee(id, emp) {

var body = JSON.stringify(emp);

var headerOptions = new Headers({ 'Content-Type': 'application/json' });

var requestOptions = new RequestOptions({ method: RequestMethod.Put, headers: headerOptions });

return this.\_http.put(this.weburl+id,body,requestOptions).map(res => res.json());}

DeletePerson(id) {

var headerOptions = new Headers({'Content-Type':'application/json'});

var requestOptions = new RequestOptions({method : RequestMethod.Delete,headers : headerOptions});

return this.\_http.delete(this.weburl+id,requestOptions).map(res => res.json());}}

**3-on the EmployeeComponent.ts**

constructor(private empService:EmployeeServiceService){

this.ShowAllPersons();}

ShowAllPersons(){

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

this.empService.Employees = data as EmployeeModule[]

);}

showForEdit(Empid:number) {

this.empService.GetPerson(Empid).subscribe(data =>

this.empService.selectedEmployee = data as EmployeeModule

);}