**Lesson17 Using Http using Http Promise**

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

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

import { headersToString } from 'selenium-webdriver/http';

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

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

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

import 'rxjs/add/operator/map';

import 'rxjs/add/operator/toPromise';

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

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

import 'rxjs/add/operator/catch';

import 'rxjs/add/observable/throw';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

import { Location } from '@angular/common';

declare var toastr:any;

@Injectable({

providedIn: 'root'

})

export class EmployeeServiceService {

constructor(private \_http:Http) { }

Employees:EmployeeModule[];

selectedEmployee:EmployeeModule;

errorMsg:string;

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

private location:Location;

//to make Fixed Loading

Page\_Load() {

location.reload();

}

//we make the following code to enable the Promise Operation

GetPersons(){

this.\_http.get(this.weburl)

.map((data : Response) =>{

return data.json() as EmployeeModule[];

}).toPromise().then(x => {

this.Employees = x;

});}

//we make the following code to enable the Promise Operation

GetPerson(id:number){

this.\_http.get(this.weburl+id)

.map((data : Response) =>{

return data.json() as EmployeeModule;

}).toPromise().then(x => {

this.selectedEmployee = x;

});}

//we make the following code to enable the Promise Operation

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()).toPromise();}

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()).toPromise();}

DeletePerson(id) {

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

}

errorHandler(error:HttpErrorResponse){

this.errorMsg = error.message;

}}

**3-in the emp-list.component.ts we will write the following code**

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

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

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

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-employee-list',

templateUrl: './employee-list.component.html',

styleUrls: ['./employee-list.component.css']

})

export class EmployeeListComponent implements OnInit {

errorMsg:string;

constructor(private empService:EmployeeServiceService,private toaster:ToastrService)

{

debugger;

this.ShowAllPersons();

}

ngOnInit() {}

ShowAllPersons()

{

this.empService.GetPersons();

this.toaster.success('Record Updated Successfully!', 'Employee Register');

}

showForEdit(Empid:number) {

this.empService.GetPerson(Empid);

}

onDelete(id: number) {

if (confirm('Are you sure to delete this record ?') == true) {

this.empService.DeletePerson(id);

//we have to make fixed loading to loading the emp-componoent.ts data

this.empService.Page\_Load();

}}}

**4-in the emp-componoent.component.ts we fill the following code**

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

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

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

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

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

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-employee',

templateUrl: './employee.component.html',

styleUrls: ['./employee.component.css']

})

export class EmployeeComponent implements OnInit {

constructor(private employeeService: EmployeeServiceService,private toaster:ToastrService) {}

private selectedEmployee:EmployeeModule;

ngOnInit() {

this.resetForm();}

resetForm(form?: NgForm) {

if (form != null)

form.reset();

this.employeeService.selectedEmployee = {

EmployeeID: null,

FirstName: '',

LastName: '',

EmpCode: '',

Position: '',

Office: ''

}}

onSubmit(form: NgForm) {

debugger;

if (form.value.EmployeeID == null) {

this.employeeService.postEmployee(form.value);

//we have to make fixed loading to loading the emp-componoent.ts data

this.employeeService.Page\_Load();

this.resetForm(form);

this.toaster.success('New Record Added Succcessfully', 'Employee Register');

}

else {

this.employeeService.putEmployee(form.value.EmployeeID, form.value);

this.resetForm(form);

//we have to make fixed loading to loading the emp-componoent.ts data

this.employeeService.Page\_Load();

this.toaster.info('Record Updated Successfully!', 'Employee Register');

}}}