**Lesson24 http error response in Angular**

**Notes:-**

**1-you can apply the error response on the http and http Client**

**\*This is the code of the Http Client And Apply The Error Message**

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

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

import { Http, Response, Headers, RequestOptions, RequestMethod }

from '@angular/http';

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

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

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

//we have to apply the catch and throw on the error Handling

import 'rxjs/add/operator/catch';

import 'rxjs/add/observable/throw';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

@Injectable({

providedIn: 'root'})

export class EmployeeService {

constructor(private \_http:HttpClient) { }

Employees:Employee[];

selectedEmployee:Employee;

weburl:string = 'http://localhost1/WebAPIProject/api/Employee/';

GetPersons() : Observable<Employee[]>{

//we apply the header sections to prepare the request

const httpOptions ={

headers: new HttpHeaders({

'accept': 'application/json'

})};

//we apply also add the catch with linking the function errorHandler

Returnthis.\_http.get<Employee[]>(this.weburl+"LoadEmployees",httpOptions).catch(this.errorHandler);}

//In this function we return the error message to the component

errorHandler(error:HttpErrorResponse){

console.log(error.message);

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

**2-on the employee-list.component.ts we write the following code**

MessageStatus:string;

constructor(public employeeService:EmployeeService){

this.MessageStatus = "Loading Employees .... Please Wait";}

ngOnInit() {

this.ShowEmployees();}

selectedEmployeeCountRadioButton: string = 'All';

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;}

GetFullName(Fname:string,Lname:string){

return Fname+Lname;}

ShowEmployees(){

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

this.employeeService.Employees = data as Employee[],

error => {

this.MessageStatus =

'Problem with the service. Please try again after sometime';

// Notice here we are logging the error to the browser console

console.error(error);

});}

**3-on the view we apply the following code**

<table>

<thead>

<tr><th colspan="4">Employee Details</th></tr>

<tr>

<td></td>

<th style="text-align: left;">Full Name</th>

<th style="text-align: left;">Gender</th>

<th style="text-align: left;">Salary</th>

</tr>

</thead>

<tbody>

<ng-container \*ngFor="let employee of employeeService.Employees;">

<tr \*ngIf="selectedEmployeeCountRadioButton=='All' ||

selectedEmployeeCountRadioButton==employee.Gender">

<td>{{count}}</td>

<td>{{GetFullName(employee.Fname,employee.Lname) | employeeTitle:employee.Gender}}</td>

<td>{{employee.Gender}}</td>

<td>{{employee.Salary | currency:'USD':true:'1.4-4'}}</td>

</tr>

</ng-container>

<tr \*ngIf="!employeeService.Employees">

<td colspan="4">

{{MessageStatus}}

</td>

</tr>

<tr \*ngIf="employeeService.Employees && employeeService.Employees.length == 0">

<td colspan="4">

No Employees to display

</td>

</tr>

</tbody>

</table>

**\*This is the code of http Module**

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

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

import {EmployeeService} from '../../Service/employee.service';

import { DELEGATE\_CTOR } from '@angular/core/src/reflection/reflection\_capabilities';

@Component({

selector: 'app-employee',

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

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

})

export class EmployeeListComponent implements OnInit {

MessageStatus:string;

constructor(public employeeService:EmployeeService){

this.MessageStatus = "Loading Employees .... Please Wait";}

ngOnInit() {

debugger;

this.ShowEmployees();}

selectedEmployeeCountRadioButton: string = 'All';

onEmployeeCountRadioButtonChange(selectedRadioButtonValue: string): void {

this.selectedEmployeeCountRadioButton = selectedRadioButtonValue;}

GetFullName(Fname:string,Lname:string){

return Fname+Lname;}

ShowEmployees(){

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

this.employeeService.Employees = data as Employee[],

error => {

this.MessageStatus =

'Problem with the service. Please try again after sometime';

// Notice here we are logging the error to the browser console

console.error(error);

});}

GetMaleEmployees():number{

return this.employeeService.Employees.filter(e => e.Gender == "Male").length;}

GetFemaleEmployees():number{

return this.employeeService.Employees.filter(e => e.Gender == "Female").length;}

GetAllEmployees():number{

return this.employeeService.Employees.length;}}

**2-on the service we apply the error handling and the Http**

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

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

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

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

//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/map';

import 'rxjs/add/operator/catch';

import 'rxjs/add/observable/throw';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

constructor(private \_http:Http) { }

Employees:Employee[];

selectedEmployee:Employee;

weburl:string = 'http://localhost1/WebAPIProject/api/Employee/';

GetPersons(): Observable<Employee[]>{

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

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

return this.\_http.get(this.weburl+'/LoadEmployees/',requestOptions).map(res => <Employee[]>res.json()).catch(this.errorHandler);}

GetPerson(id:number) : Observable<Employee>{

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

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

return this.\_http.get(this.weburl+'/LoadEmployees/'+id,requestOptions).map(res => <Employee>res.json()).catch(this.errorHandler);}

GetPersonBasedOnGender(Gender:string) : Observable<Employee[]>{

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

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

return this.\_http.get(this.weburl+"LoadEmployees?Gender=",requestOptions).map(res => <Employee[]>res.json()).catch(this.errorHandler);}

errorHandler(error:Response){

console.log(error);

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

}}