**Lesson14 for Loop in Angular**

**Notes:-**

**1-ngFor is usually used to display an array of items**

**2-Since ngFor is a structural directive it is prefixed with \***

**3-\*ngFor='let employee of employees' - In this example 'employee' is called template input variable, which can be accessed by the <tr> element and any of it's child elements.**

**4-ngIf structural directive displays the row "No employees to display" when employees property does not exist or when there are ZERO employees in the array.**

**Example**

**In this example we will make array of Employee as the following**

**1-on the Employee.ts we type the below code**

export class Employee

{

ID:number;

Fname:string;

Lname:string;

Gender:string;

Salary:number;

}

**2-on the Employee-service.ts we tye the below code**

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

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

import 'rxjs/add/observable/throw';

//we put observable in the service.ts

import { Observable } from 'rxjs/Observable';

declare var toastr:any;

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

constructor(private \_http:HttpClient) { }

Employees:Employee[];

selectedEmployee:Employee;

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

GetPersons() : Observable<Employee[]>{

debugger

const httpOptions ={

headers: new HttpHeaders({

'accept': 'application/json'

})};

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

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

const httpOptions ={

headers: new HttpHeaders({

'accept': 'application/json'

})};

Returnthis.\_http.get<Employee>(this.weburl+"LoadEmployee/"+id,httpOptions)

.catch(this.errorHandler);}

postEmployee(emp : Employee){

debugger;

var body = JSON.stringify(emp);

const httpOptions ={

headers: new HttpHeaders({

'accept': 'application/json',

'Content-Type': 'application/json'

})};

Returnthis.\_http.post<Employee>(this.weburl,body,httpOptions).catch(this.errorHandler);}

putEmployee(id, emp) {

debugger;

var body = JSON.stringify(emp);

const httpOptions ={

headers: new HttpHeaders({

'Content-Type': 'application/json',

'Authorization': 'my-auth-token'

})};

returnthis.\_http.put<Employee>(this.weburl+"/"+id,body,httpOptions).catch(this.errorHandler);}

DeletePerson(id) {

const httpOptions ={

headers: new HttpHeaders({

'Content-Type': 'application/json',

'Authorization': 'my-auth-token'

})};

Return this.\_http.delete<Employee>(this.weburl+"/"+id,httpOptions).

catch(this.errorHandler);}

errorHandler(error:HttpErrorResponse){

debugger;

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

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

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: './employee.component.html',

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

})

export class EmployeeComponent implements OnInit {

constructor(public employeeService:EmployeeService) { }

public columnSpan:number;

public ShowHide:boolean;

public lst:Employee[];

public FullName:string;

ngOnInit() {

this.ShowAllPersons();

this.ShowHide = true;

this.columnSpan = 4;

this.FullName = "TOM";}

Change(FullNameVar){

this.FullName = FullNameVar;}

ShowAllPersons(){

debugger;

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

this.employeeService.Employees = data as Employee[]);}

ShowHideEvent(){

this.ShowHide = !this.ShowHide;}}

**3-on the view , we type the following code**

<table>

<thead>

<tr>

<th attr.colspan="{{columnSpan}}">Employee Details</th>

</tr>

<tr>

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

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

<th \*ngIf="ShowHide" style="text-align: left;">Gender</th>

<th \*ngIf="ShowHide" style="text-align: left;">Salary</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let item of employeeService.Employees">

<td>{{item.Fname}}</td>

<td>{{item.Lname}}</td>

<td \*ngIf="ShowHide" >{{item.Gender}}</td>

<td \*ngIf="ShowHide">{{item.Salary}}</td>

</tr>

//to Check that the Employee[] variable have the objects

<tr \*ngIf="employeeService.Employees==0|| !employeeService.Employees">

<td colspan="4">

No Employees to display

</td>

</tr>

</tbody>

</table>

<button (click)="ShowHideEvent(ButtonState)">

{{ShowHide ? 'Hide' : 'Show'}} Button</button>