**Angular 9 application => CURD Operation**

1. Download latest version of **node js**
2. Install angular cli latest version -> **npm install –g @angular/cli@latest**
3. App.module.ts

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

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

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

import { AppRoutingModule } from './app-routing.module';

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

import { HomeModule } from '../app/home/home.module';

@NgModule({

declarations: [

AppComponent

],

imports: [

BrowserModule,

HttpClientModule,

AppRoutingModule,

HomeModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

==============================================================================

1. App-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

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

import { HomeComponent } from './home/home.component';

const routes: Routes = [

{ path: '', component: AppComponent },

{ path: 'login', component: AppComponent },

{

path: 'home',

loadChildren: () => import('../app/home/home.module').then(m => m.HomeModule)

}

];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

===================================================================================

1. App.comoponent.html

<router-outlet></router-outlet>

===================================================================================

1. App.component.ts

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

@Component({

selector: 'app-root',

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

styleUrls: ['./app.component.scss']

})

export class AppComponent {

title = 'ClientApp';

}

===================================================================================

1. [Home.module.ts](http://Home.module.ts)

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

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

import { FormsModule } from '@angular/forms'

import { HomeComponent } from '../home/home.component';

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

import { HomeRoutingModule } from '../../app/home/home-routing.module';

//import { MatFormFieldModule } from '@angular/material/form-field';

//import { MatInputModule } from '@angular/material/input';

@NgModule({

declarations: [

HomeComponent,

],

imports: [

CommonModule,

FormsModule,

HomeRoutingModule,

// MatFormFieldModule,

// MatInputModule

],

providers: [

EmployeeService

],

bootstrap: [HomeComponent]

})

export class HomeModule { }

===================================================================================

1. Home-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

import { HomeComponent } from '../home/home.component';

const routes: Routes = [

{

path: '', component: HomeComponent,

children: [

{

path: 'employee', loadChildren: () => import('../home/employee/employee.module').then(m => m.EmployeeModule)

}

]

},

];

@NgModule({

imports: [RouterModule.forChild(routes)],

exports: [RouterModule]

})

export class HomeRoutingModule { }

===================================================================================

1. [Home.component.html](http://Home.component.html)

<div class="row">

<div class="row col-md-12">

<div class="col-md-2" style="background:#0041b3">

<ul>

<li style="list-style-type: none;">

<a style="color:white;" routerLink="/employee">Employee</a>

</li>

</ul>

</div>

<div class="col-md-10">

<router-outlet></router-outlet>

</div>

</div>

</div>

===================================================================================

1. [Home.component.ts](http://Home.component.ts)

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

@Component({

selector: 'app-home',

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

styleUrls: ['./home.component.scss']

})

export class HomeComponent implements OnInit {

constructor() { }

ngOnInit(): void {

}

}

===================================================================================

1. Employee.model.ts

export class EmployeeModel {

public empId: number;

public empName: string;

public empEmailId: string;

public empaddress: string;

}

===================================================================================

1. Employee.module.ts

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

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

import { FormsModule } from '@angular/forms'

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

import { EmployeeRoutingModule } from '../employee/employee-routing.module';

@NgModule({

declarations: [

EmployeeComponent

],

imports: [

CommonModule,

FormsModule,

EmployeeRoutingModule

],

providers: [],

bootstrap: [EmployeeComponent]

})

export class EmployeeModule { }

===================================================================================

1. Employee-routing.module.ts

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

import { RouterModule, Routes } from '@angular/router';

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

const routes: Routes = [

{ path: '', component: EmployeeComponent },

];

@NgModule({

imports: [RouterModule.forChild(routes)],

exports: [RouterModule]

})

export class EmployeeRoutingModule { }

===================================================================================

1. Employee.component.html

<div class="col-md-12" \*ngIf="!isCreateView">

<div class="col-md-12">

<button type="button" name="btnCreate" (click)="createEmployee()">Create</button>

</div>

<div class="col-md-12">

<table>

<thead>

<tr>

<th>

Action

</th>

<th>Employee Name</th>

<th>Email Id</th>

<th>Address</th>

</tr>

</thead>

<tbody>

<tr \*ngFor="let emp of empList" (click)="editData(emp)">

<td>

<button (click)="deleteEmployee(emp)">Delete</button>

</td>

<td>**{{**emp.empName**}}**</td>

<td>**{{**emp.empEmailId**}}**</td>

<td>**{{**emp.empaddress**}}**</td>

</tr>

</tbody>

</table>

</div>

</div>

<div class="col-md-12" \*ngIf="isCreateView">

<form name="employeeForm">

<div class="col-md-12">

<div class="col-md-4">

<span>Employee Name</span>

</div>

<div class="col-md-4">

<input type="text" id="empname" name="empname"

[(ngModel)]="employee.empName" />

</div>

</div>

<div class="col-md-12">

<div class="col-md-4">

<span>Email Id</span>

</div>

<div class="col-md-4">

<input type="text" id="empEmailId" name="empEmailId"

[(ngModel)]="employee.empEmailId" />

</div>

</div>

<div class="col-md-12">

<div class="col-md-4">

<span>Address</span>

</div>

<div class="col-md-4">

<input type="text" id="empAddress" name="empAddress"

[(ngModel)]="employee.empaddress" />

</div>

</div>

<div class="col-md-12">

<button id="btnSave" (click)="saveEmployee()">Save</button>

<button id="btnSave" (click)="cancel()">Cancel</button>

</div>

</form>

</div>

===================================================================================

1. Employee.component.ts

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

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

import { EmployeeModel } from '../employee/models/employee.model';

@Component({

selector: 'app-employee',

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

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

})

export class EmployeeComponent implements OnInit {

public empList: EmployeeModel[] = [];

public employee: EmployeeModel = new EmployeeModel();

public isCreateView: boolean = false;

constructor(private employeeService: EmployeeService) { }

ngOnInit(): void {

this.getEmployeeList();

}

getEmployeeList() {

this.isCreateView = false;

this.employeeService.getEmployeelist().subscribe((res: any) => {

this.empList = res;

}, (error) => {

console.log(error);

});

}

createEmployee() {

this.isCreateView = true;

this.employee = new EmployeeModel();

}

editData(emp: any) {

this.isCreateView = true;

this.employee = new EmployeeModel();

this.employee = emp;

}

saveEmployee() {

if (this.employee.empId == undefined || this.employee.empId <= 0) {

this.employee.empId = 0;

this.addEmployee(this.employee);

}

else {

this.updateEmployee(this.employee);

}

}

addEmployee(employee: EmployeeModel) {

this.employeeService.saveEmployee(employee).subscribe((res: any) => {

if (res > 0) {

alert("Record Saved");

}

this.getEmployeeList();

this.isCreateView = false;

}, (error) => {

console.log(error);

});

}

updateEmployee(employee: EmployeeModel) {

this.employeeService.updateEmployee(employee).subscribe((res: any) => {

if (res > 0) {

alert("Record Updated");

}

this.getEmployeeList();

this.isCreateView = false;

}, (error) => {

console.log(error);

});

}

deleteEmployee(emp: EmployeeModel) {

this.isCreateView = false;

this.employeeService.deleteEmployee(emp.empId).subscribe((res: any) => {

if (res > 0) {

alert("Record Deleted");

}

this.getEmployeeList();

}, (error) => {

console.log(error);

});

}

cancel() {

this.isCreateView = false;

}

}

==============================================================================

1. Employee.service.ts

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

import { Observable, of, throwError } from 'rxjs';

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

import { catchError, tap, map } from 'rxjs/operators';

import { EmployeeModel } from '../home/employee/models/employee.model';

const apiServer = "http://localhost:5744/api/Employee/";

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

httpOptions = {

headers: new HttpHeaders({

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

})

}

constructor(private http: HttpClient) { }

getEmployeelist(): Observable<any> {

const url = apiServer + "getEmployeeList";

return this.http.get(url,this.httpOptions).pipe(

map(response => {

return response;

}),

catchError(error => {

return error;

})

);

}

getEmployeeById(id: number): Observable<any> {

const url = apiServer + "getEmployeeById/" + id;

return this.http.get(url, this.httpOptions).pipe(

map(response => {

return response;

}),

catchError(error => {

return error;

})

);

}

saveEmployee(employee: EmployeeModel): Observable<any> {

const url = apiServer + "saveEmployee";

return this.http.post<any>(url, JSON.stringify(employee), this.httpOptions).pipe(

map(response => {

return response;

}),

catchError(error => {

return error;

})

);

}

updateEmployee(employee: EmployeeModel): Observable<any> {

const url = apiServer + "updateEmployee";

return this.http.put<any>(url, JSON.stringify(employee), this.httpOptions).pipe(

map(response => {

return response;

}),

catchError(error => {

return error;

})

);

}

deleteEmployee(id: number): Observable<any> {

const url = apiServer + "deleteEmployee/" + id;

return this.http.delete(url).pipe(

map(response => {

return response;

}),

catchError(error => {

return error;

})

);

}

}