**Hands-On: Stage 3 - Angular SPA-021 to SPA-023 - Day 71 – Hands on 4**

**Session - 1**

1. **Component Interaction - Create Component and Display Employee List**

**emp-info.component.ts**

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

import { ActivatedRoute } from '@angular/router';

@Component({

  selector: 'app-emp-info',

  templateUrl: './emp-info.component.html'

})

export class EmpInfoComponent {

  empObj :any={};

  @Input() employee = [{id:1, name:"John", salary:5000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:1, name:"HTML"}] },

  {id:2, name:"Smith", salary:10000, permanent:true, department:[{id:6, name:"Internal"}],skills:[{id:2, name:"JAVA"}] },

  {id:3, name:"Mary", salary:55000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] },

  {id:4, name:"Sandy", salary:15000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:3, name:"CSS"}] },

  {id:5, name:"Martin", salary:20000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] }

]

  constructor(private route:ActivatedRoute)

  {

      let eno  =  this.route.snapshot.params["id"];

      this.empObj = this.employee.find(  item => item.id == eno );

  }

}

interface Department{

  id:number;

  name:string;

}

 interface Skill{

  id:number;

  name:string;

}

 interface Employee extends Department, Skill{

  id:number;

  name:string;

  salary:number;

  permanent:boolean;

  department:Department[];

  skills:Skill[];

}

**emp-info.component.html**

<h2>Employee Details</h2>

<div class="cardblock">

    <br>

    Name : {{empObj.name}}

    <br>

    <br>

    Salary : {{empObj.salary | currency : "INR"}}

    <br>

    <br>

    Permanent : {{empObj.permanent}}

    <br>

    <br>

    <a class="editLink" [routerLink]="['/edit-emp-reactive',empObj.id]">Edit</a>

</div>

**employee-list.component.html**

<h3>Employee List</h3>

<div style="margin: 20px;">

    Search &nbsp;

    <input type="text" placeholder="Find by name" [(ngModel)]="searchKey"  (input)="Search()">

</div>

<div class="row">

<div class="container" \*ngFor="let emp of employeeList">

<div class="column" style="padding:5px;">

    <div class="cardblock" style="width: 18rem;border: 2px solid #000000;border-color:#0645AD;padding:5px;">

        <span style="font-size: 30px;">{{emp.name}}</span>

        <span style="float:right ;"> <a style="color: red;" [routerLink]="['/emp-info',emp.id]">emp-info</a></span>

        <br>

        {{emp.salary | currency : "INR"}}<br>

        <a class="editLink" [routerLink]="['/edit-emp-reactive',emp.id]">Edit</a>

    </div>

</div>

</div>

</div>

**employee-list.component.ts**

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

import { ActivatedRoute } from "@angular/router";

@Component({

  selector: 'app-employee-list',

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

})

export class EmployeeListComponent implements OnInit{

   employeeList:Employee[]=[];

ngOnInit(){

  this.employeeList = [{id:1, name:"John", salary:5000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:1, name:"HTML"}] },

  {id:2, name:"Smith", salary:10000, permanent:true, department:[{id:6, name:"Internal"}],skills:[{id:2, name:"JAVA"}] },

  {id:3, name:"Mary", salary:55000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] },

  {id:4, name:"Sandy", salary:15000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:3, name:"CSS"}] },

  {id:5, name:"Martin", salary:20000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] }

]

}

public searchKey:string="";

Search()

{

  if(this.searchKey != "")

  {

    this.employeeList = this.employeeList.filter(res=>{

      return res.name.toLocaleLowerCase().match(this.searchKey.toLocaleLowerCase());

});

  }

  else if(this.searchKey == "")

  {

    this.ngOnInit();

  }

}

}

 interface Department{

  id:number;

  name:string;

}

 interface Skill{

  id:number;

  name:string;

}

 interface Employee extends Department, Skill{

  id:number;

  name:string;

  salary:number;

  permanent:boolean;

  department:Department[];

  skills:Skill[];

}

**app.module.ts**

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

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

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

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

import { ViewempsComponent } from './viewemps/viewemps.component';

import { EditEmpReactiveComponent } from './editemp/edit-emp-reactive.component';

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

import { EmpInfoComponent } from './emp-info/emp-info.component';

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

import { viewClassName } from '@angular/compiler';

import { QuantityIncrementComponent } from './quantity-increment/quantity-increment.component';

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

const appRoutes:Routes=[{path:"" , component:ViewempsComponent},

  {path:"EditEmployees" ,

  component:EditEmpReactiveComponent},{path:"QuantityIncrement",component:QuantityIncrementComponent},

  { path: 'edit-emp-reactive/:id', component: EmployeeListComponent}]

@NgModule({

  declarations: [

    AppComponent,

    ViewempsComponent,

    EmployeeListComponent,

    EmpInfoComponent,

    EditEmpReactiveComponent,

    QuantityIncrementComponent,

  ],

  imports: [

    FormsModule,

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    RouterModule.forRoot(appRoutes)

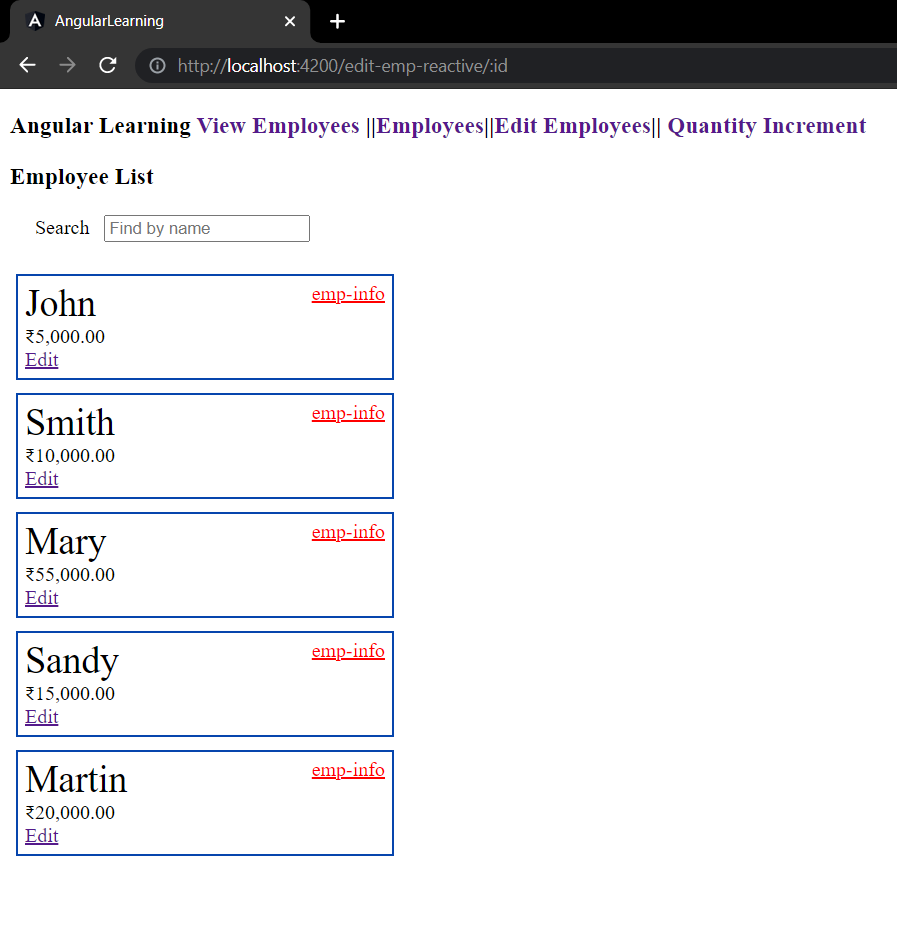
  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }

**OUTPUT:**

**Session - 2**

1. **Service - Create service for handling employee data**

**employee.service.ts**

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

@Injectable({

  providedIn: 'root'

})

export class EmployeeService {

  employeeList:Employee[]= [{id:1, name:"John", salary:5000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:1, name:"HTML"}] },

  {id:2, name:"Smith", salary:10000, permanent:true, department:[{id:6, name:"Internal"}],skills:[{id:2, name:"JAVA"}] },

  {id:3, name:"Mary", salary:55000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] },

  {id:4, name:"Sandy", salary:15000, permanent:true, department:[{id:3, name:"Payroll"}],skills:[{id:3, name:"CSS"}] },

  {id:5, name:"Martin", salary:20000, permanent:true, department:[{id:2, name:"Lead"}],skills:[{id:1, name:"HTML"}] }]

  getAllEmployees() : Employee[]{

      return this.employeeList;

  }

  getEmployee(employeeId: number){

    return this.employeeList.filter(e=>e.id===employeeId);

  }

}

export interface Department{

  id:number;

  name:string;

}

 interface Skill{

  id:number;

  name:string;

}

 interface Employee extends Department, Skill{

  id:number;

  name:string;

  salary:number;

  permanent:boolean;

  department:Department[];

  skills:Skill[];

}

**auth.service.ts**

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

@Injectable({

  providedIn: 'root'

})

export class AuthService {

  constructor() { }

  loggedIn!: boolean;

  login()

  {

    this.loggedIn=true;

  }

  logout()

  {

    this.loggedIn=false;

  }

  isLoggedIn():boolean

  {

    return this.loggedIn;

  }

}

**auth.guard.ts**

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

import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot, UrlTree } from '@angular/router';

import { Observable } from 'rxjs';

import { AuthService } from '../auth.service';

import { Router, ActivatedRoute, ParamMap } from '@angular/router';

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  constructor(private authService: AuthService,

    private router: Router) { }

    canActivate(

      next: ActivatedRouteSnapshot,

      state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {

        if (this.authService.isLoggedIn()) {

          return true;

           }

           else{

            this.router.navigate(['/login']);

           }

        return false;

    }}