**SERVICES AND GUARDS**

**Question 1**

**Service - Create service for handling employee data**  
  
Follow steps below the create the service and utilize necessary methods edit-employee-reactive component:

* Create a new service using following command. Execute this command in the root folder of the application.

ng generate service employee

* Create a property named employeeList as array of Employee type.
* Copy and paste the hardcoded employeeList from employee-list component.
* Implement the below methods in the service class.
  + getAllEmployees() : Employee[]
    - This method will return the employeeList property
  + getEmployee(employeeId: number): Employee
    - This method will iterate through employeeList and return based on the matching employeeId
* To use this service edit-employee-reactive component by changing the constructor as following, which will inject the service into this component. Include necessary import for EmployeeService.

constructor(private route : ActivatedRoute, private employeeService : EmployeeService) { }

* Assign the employee property of edit-employee-reactive component using employeeService.getEmployee() method passing the employee id obtained from the router link.

**Answer 1**

*Rest all same as previous*

**employee.service.ts**

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

import { Employee } from './Employee';

import { empty } from 'rxjs';

@**Injectable**({

providedIn: 'root'

})

export class **EmployeeService** {

constructor() { }

employeeList:**Employee**[]=[{

id:1,

name:'John',

salary:10000,

permanent:true,

skills:null,

department:null,

dateofBirth:null

},

{

id:2,

name:'Dona',

salary:980000,

permanent:true,

dateofBirth:null,

skills:null,

department:null

},

{

id:3,

name:'Jasmine',

salary:90000,

permanent:true,

dateofBirth:null,

skills:null,

department:null

},

{

id:4,

name:'Akash',

salary:80000,

permanent:true,

dateofBirth:null,

skills:null,

department:null

}

];

**getAllEmployees**():**Employee**[]

{

return this.employeeList

}

**getEmployee**(employeeId: number): **Employee**

{

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

}

}

**edit-emp-reactive.component.ts**

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

import { FormGroup, FormControl, Validators } from '@angular/forms';

import { Department } from '../Department';

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

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

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

@**Component**({

selector: 'app-edit-emp-reactive',

templateUrl: './edit-emp-reactive.component.html',

styleUrls: ['./edit-emp-reactive.component.css']

})

export class **EditEmpReactiveComponent** implements **OnInit** {

employee:**Employee**;

reactiveForm: **FormGroup**;

public employees:**Employee**[]

constructor(private route:**ActivatedRoute**,private empService:**EmployeeService**) {

this.employees=empService.**getAllEmployees**();

}

 departments: **Department**[]=[

    {id:1,name:"HelpDesk"},

    {id:2,name:"HR"},

    {id:3,name:"IT"},

    {id:4,name:"Payroll"},

    {id:5,name:"admin"}

  ];

  emp:**Employee**={

    id:null,

    name:null,

    salary:null,

    permanent:null,

    department:{

      id:1,

      name:'Payroll'

    },

    skills:[

      {

        id:1,

        name:'HTML'

      },

      {

        id:2,

        name:'CSS'

      },

      {

        id:3,

        name:'Javascript'

      }

    ],

    dateofBirth: new **Date**('04/20/2019')

}

**ngOnInit**() {

this.**initForm**();

}

get **empName**() {

return this.reactiveForm.**get**('empName');

}

**onSubmit**() {

**console**.**log**("Name is : " + this.reactiveForm.value.empName);

**console**.**log**("Salary is : " + this.reactiveForm.value.empSalary);

**console**.**log**("Permananent or Not : " + this.reactiveForm.value.empPermananent);

**console**.**log**("Department is : " + this.reactiveForm.value.empDept);

}

private **initForm**() {

const eid = +this.route.snapshot.paramMap.**get**('id');

this.employee=this.empService.**getEmployee**(eid);

this.reactiveForm = new **FormGroup**({

'empName': new **FormControl**(this.employee.name,[

Validators.required,

Validators.**minLength**(4),

Validators.**maxLength**(20)

]),

'empSalary':new **FormControl**(this.employee.salary),

'empPermananent':new **FormControl**(this.employee.permanent),

'empDept':new **FormControl**(this.employee.department)

}

);

}

}

**Question 2**

**Guard - Login Component**  
  
Restrict access to edit-employee-reactive component with a login page.  
  
The credentials can be hardcode to 'admin'/'password' for the sake of simplicity.  
  
We will implement guard feature gradually.  
  
In this hands on we will create a login component. Follow instructions below:

* Create new component with name 'login'
* Include properties username and password
* Include form in HTML template with following fields:
  + Username - textbox - include required validation
  + Password - password - include required validation
  + Login - Submit Button pointing to login() method in login component.
* Include code in login() to verify if username is 'admin' and password is 'password', if so navigate to edit-emp-reactive route. The status of successful login needs to be stored in a service. This will be done in next hands on.
* If the credentials does not match display message 'Invalid Username/Password' within the login component.
* Include a new header navigation link named Login and bind a path in router module

**Answer 2**

**app-routing.module.ts**

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

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

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

import { QuantitySelectorComponent } from './quantity-selector/quantity-selector.component';

import { EditEmpComponent } from './edit-emp/edit-emp.component';

import { ViewEmpComponent } from './view-emp/view-emp.component';

import { EditEmpTemplateDrivenComponent } from './edit-emp-template-driven/edit-emp-template-driven.component';

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

import { EmployeesComponent } from './employees/employees.component';

import { LoginComponent } from './login/login.component';

const routes: **Routes** = [

{

path:'view',

component:ViewEmpComponent

},

{

path:'edit',

component:EditEmpComponent

},

{

path:'quant-incr',

component:QuantityIncrementComponent

},

{

path:'quant-sel',

component:QuantitySelectorComponent

},

{

path:'edit-emp-temp',

component:EditEmpTemplateDrivenComponent

},

{

path:'edit-emp-react/:id',

component:EditEmpReactiveComponent

},

{

path:'employees-list',

component:EmployeesComponent

},

{

path:'login',

component:LoginComponent

}

];

@**NgModule**({

imports: [RouterModule.**forRoot**(routes)],

exports: [RouterModule]

})

export class **AppRoutingModule** { }

**login.component.ts**

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

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

@**Component**({

selector: 'app-login',

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

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

})

export class **LoginComponent** implements **OnInit** {

msg:string;

username:string=null;

password:string=null;

constructor(private \_router: **Router**) { }

**ngOnInit**() {

}

**login** (){

if(this.username=="admin" && this.password=="password")

{

this.\_router.**navigate**(['edit-emp-react/1']);

}

else

{

this.msg="Invalid User Name and Password"

this.\_router.**navigate**(['login'])

}

}

}

**login.component.html**

<form *(ngSubmit)*="login()" *#loginForm*="ngForm">

<div>

<label>Username</label>&nbsp;

<input *type*="text" *name*="username" *[(ngModel)]*="username" *required*>

</div>

<div>

<label>Password</label>&nbsp;

<input *type*="password" *name*="password" *[(ngModel)]*="password" *required*>

</div>

<div>

<button *type*="submit" *[disabled]*="!loginForm.valid">Login</button>

</div>

{{msg}}

</form>

**Question 3**

**Guard - AuthService**  
  
Implement AuthService that will be used to store and retrieve the authentication status. Follow steps below to implement and use it:

* Create new service named 'auth'
* Include property loggedIn of type boolean.
* Implement method login() that sets the loggedIn property to true.
* Implement method logout() that sets the loggedIn property to false.
* Implement method isLoggedIn() that returns the value of loggedIn property
* Inject this service in login component constructor
* Invoke authService.login() when Username and Password matches to 'admin'/'password'

​​​​​​​**Answer 3**

**auth.service.ts**

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

@**Injectable**({

providedIn: 'root'

})

export class **AuthService** {

loggedIn=false;

**login**()

{

this.loggedIn=true;

}

**logout**()

{

this.loggedIn=false;

}

**isloggedIn**():boolean

{

return this.loggedIn;

}

constructor() { }

}

**login.component.ts**

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

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

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

@**Component**({

selector: 'app-login',

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

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

})

export class **LoginComponent** implements **OnInit** {

msg:string;

username:string=null;

password:string=null;

constructor(private \_router: **Router**,private authServie:**AuthService**) { }

**ngOnInit**() {

}

**onSubmit**(){

if(this.username=="admin" && this.password=="password")

{

this.authServie.**login**();

this.\_router.**navigate**(['edit-emp-react/1']);

}

else

{

this.msg="Invalid User Name and Password"

this.\_router.**navigate**(['login'])

}

}

}

**Question 4**

**Guard - Protect Component using Guard**  
  
Only admins will be able to edit an employee detail. The edit-employee-reactive form needs to be available only after login. If this form is accessed through URL or router navigation link without login, then it should be redirected to login component.  
  
Steps to implement:

* Create a new guard with name auth

ng generate guard auth/auth

* Inject Router and AuthService in the constructor
* Implement canActivate() method that returns boolean
* If login status is true from authService return true else navigate to login component using router.navigate() method.
* In app-routing.module.ts include canActivate for edit-employee-reactive path with reference to AuthGuardService as guard.

**Answer 4**

*Rest all same as previous*

**auth-guard.service.ts**

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

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

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

@**Injectable**({

providedIn: 'root'

})

export class **AuthGuardService** implements **CanActivate** {

isLoggedIn:boolean;

**canActivate**(route: import("@angular/router").**ActivatedRouteSnapshot**, state: import("@angular/router").**RouterStateSnapshot**): boolean | import("@angular/router").**UrlTree** | import("rxjs").**Observable**<boolean | import("@angular/router").**UrlTree**> | **Promise**<boolean | import("@angular/router").**UrlTree**> {

this.isLoggedIn = this.\_authService.loggedIn;

**console**.**log**("isLoggedIn: " + this.isLoggedIn);

if(this.isLoggedIn){

return true;

}

else{

this.\_router.**navigate**(['login']);

return false;

}

}

constructor(private \_router : **Router**, private \_authService :**AuthService**) {}

}

**app-routing.module.ts**

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

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

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

import { QuantitySelectorComponent } from './quantity-selector/quantity-selector.component';

import { EditEmpComponent } from './edit-emp/edit-emp.component';

import { ViewEmpComponent } from './view-emp/view-emp.component';

import { EditEmpTemplateDrivenComponent } from './edit-emp-template-driven/edit-emp-template-driven.component';

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

import { EmployeesComponent } from './employees/employees.component';

import { LoginComponent } from './login/login.component';

import { AuthGuardService } from './auth-guard.service';

const routes: **Routes** = [

{

path:'view',

component:ViewEmpComponent

},

{

path:'edit',

component:EditEmpComponent

},

{

path:'quant-incr',

component:QuantityIncrementComponent

},

{

path:'quant-sel',

component:QuantitySelectorComponent

},

{

path:'edit-emp-temp',

component:EditEmpTemplateDrivenComponent

},

{

path:'edit-emp-react/:id',

component:EditEmpReactiveComponent,

canActivate: [AuthGuardService]

},

{

path:'employees-list',

component:EmployeesComponent

},

{

path:'login',

component:LoginComponent

}

];

@**NgModule**({

imports: [RouterModule.**forRoot**(routes)],

exports: [RouterModule]

})

export class **AppRoutingModule** { }