Ng g g guards/auth

Before we use this guard, we must inject it into app.module.ts file

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

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

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  canActivate(

    route: ActivatedRouteSnapshot,

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

    return true;

  }

}

import {AuthGuard} from './guard/auth.guard';

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

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

import { DepartmentContactComponent } from './department-contact/department-contact.component';

import { DepartmentDetailComponent } from './department-detail/department-detail.component';

import { DepartmentOverviewComponent } from './department-overview/department-overview.component';

import { DepartmentsComponent } from './departments/departments.component';

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

import { NotFoundComponent } from './not-found/not-found.component';

import {AuthGuard} from './guard/auth.guard';

const routes: Routes = [

{path:'' , redirectTo: '/departments' , pathMatch:'full'},

{path:'departments' ,component: DepartmentsComponent},

{path:'departments/:id' ,component: DepartmentDetailComponent,

children:

[

{path:'overview', component:DepartmentOverviewComponent},

{path:'contact', component: DepartmentContactComponent}

]

},

{path:'employees' , component: EmployeesComponent,

canActivate: [AuthGuard]},

{path:"\*\*" ,  component: NotFoundComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule],

  providers:[AuthGuard]

})

export class AppRoutingModule { }

export const routingComponents =[DepartmentsComponent,

  EmployeesComponent, NotFoundComponent,

  DepartmentDetailComponent,DepartmentOverviewComponent

,DepartmentContactComponent];

We can also create a service

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

@Injectable({

  providedIn: 'root'

})

export class UserService {

  constructor() { }

  isAdminRights() : boolean

  {

    return  true;

  }

}

Import it in app.module.ts file

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

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

import { nextTick } from 'process';

import { Observable } from 'rxjs';

import {UserService} from '../user.service';

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  constructor(private \_user: UserService){}

  user = {

    role :'admin'

  }

  canActivate(

    route: ActivatedRouteSnapshot,

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

    if(this.\_user.isAdminRights()==true)

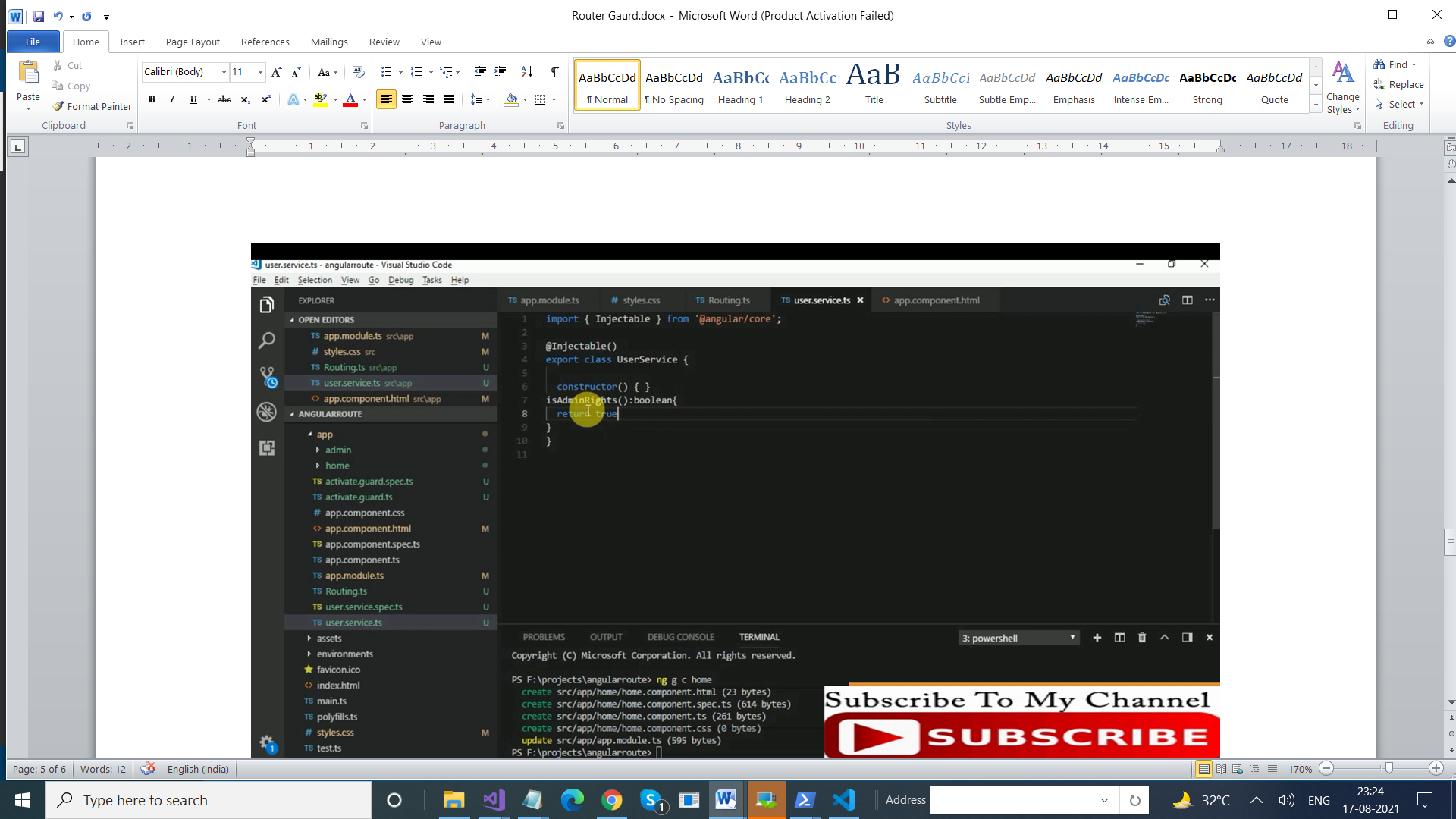
    return true;

    else

    return false;

  }

}



Role based

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

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

import { nextTick } from 'process';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  user = {

    role :'admin'

  }

  canActivate(

    route: ActivatedRouteSnapshot,

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

    if(route.data[0]==this.user.role)

      return true;

      else

      return false;

  }

}

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

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

import { DepartmentContactComponent } from './department-contact/department-contact.component';

import { DepartmentDetailComponent } from './department-detail/department-detail.component';

import { DepartmentOverviewComponent } from './department-overview/department-overview.component';

import { DepartmentsComponent } from './departments/departments.component';

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

import { NotFoundComponent } from './not-found/not-found.component';

import {AuthGuard} from './guard/auth.guard';

const routes: Routes = [

{path:'' , redirectTo: '/departments' , pathMatch:'full'},

{path:'departments' ,component: DepartmentsComponent},

{path:'departments/:id' ,component: DepartmentDetailComponent,

children:

[

{path:'overview', component:DepartmentOverviewComponent},

{path:'contact', component: DepartmentContactComponent}

]

},

{path:'employees' , component: EmployeesComponent,

canActivate: [AuthGuard], data:['admin']},

{path:"\*\*" ,  component: NotFoundComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule],

  providers:[AuthGuard]

})

export class AppRoutingModule { }

export const routingComponents =[DepartmentsComponent,

  EmployeesComponent, NotFoundComponent,

  DepartmentDetailComponent,DepartmentOverviewComponent

,DepartmentContactComponent];

CanActivateChild

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

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

import { nextTick } from 'process';

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class AuthGuard implements CanActivate {

  user = {

    role :'admin'

  }

  canActivate(

    route: ActivatedRouteSnapshot,

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

    if(route.data[0]==this.user.role)

      return true;

      else

      return false;

  }

}

export class AuthGuard1 implements CanActivateChild {

  user = {

    role :'admin'

  }

  canActivateChild(

    route: ActivatedRouteSnapshot,

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

    if(route.data[0]==this.user.role)

      return true;

      else

      return false;

  }

}

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

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

import { DepartmentContactComponent } from './department-contact/department-contact.component';

import { DepartmentDetailComponent } from './department-detail/department-detail.component';

import { DepartmentOverviewComponent } from './department-overview/department-overview.component';

import { DepartmentsComponent } from './departments/departments.component';

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

import { NotFoundComponent } from './not-found/not-found.component';

import {AuthGuard, AuthGuard1} from './guard/auth.guard';

const routes: Routes = [

{path:'' , redirectTo: '/departments' , pathMatch:'full'},

{path:'departments' ,component: DepartmentsComponent},

{path:'departments/:id' ,component: DepartmentDetailComponent,

canActivateChild: [AuthGuard1],

children:

[

{path:'overview', component:DepartmentOverviewComponent,data:['admin']},

{path:'contact', component: DepartmentContactComponent,data:['admin']}

]

},

{path:'employees' , component: EmployeesComponent,

},

{path:"\*\*" ,  component: NotFoundComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule],

  providers:[AuthGuard,AuthGuard1]

})

export class AppRoutingModule { }

export const routingComponents =[DepartmentsComponent,

  EmployeesComponent, NotFoundComponent,

  DepartmentDetailComponent,DepartmentOverviewComponent

,DepartmentContactComponent];

Deactivate

Ng g g guard/check

Add a component

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

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

import { Observable } from 'rxjs';

import { TestComponent } from '../test/test.component';

@Injectable({

  providedIn: 'root'

})

export class CheckGuard implements CanDeactivate<TestComponent> {

  canDeactivate(

    test : TestComponent,

    route: ActivatedRouteSnapshot,

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

    return true;

  }

}

Inject this in module

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

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

import { DepartmentContactComponent } from './department-contact/department-contact.component';

import { DepartmentDetailComponent } from './department-detail/department-detail.component';

import { DepartmentOverviewComponent } from './department-overview/department-overview.component';

import { DepartmentsComponent } from './departments/departments.component';

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

import { NotFoundComponent } from './not-found/not-found.component';

import {AuthGuard, AuthGuard1} from './guard/auth.guard';

import { CheckGuard } from './guard/check.guard';

import { TestComponent } from './test/test.component';

const routes: Routes = [

{path:'' , redirectTo: '/departments' , pathMatch:'full'},

{path:'departments' ,component: DepartmentsComponent},

{path:'test' ,component: TestComponent ,

canDeactivate: [CheckGuard] },

{path:'departments/:id' ,component: DepartmentDetailComponent,

canActivateChild: [AuthGuard1],

children:

[

{path:'overview', component:DepartmentOverviewComponent,data:['admin']},

{path:'contact', component: DepartmentContactComponent,data:['admin']}

]

},

{path:'employees' , component: EmployeesComponent,

},

{path:"\*\*" ,  component: NotFoundComponent}

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule],

  providers:[AuthGuard,AuthGuard1,CheckGuard]

})

export class AppRoutingModule { }

export const routingComponents =[DepartmentsComponent,

  EmployeesComponent, NotFoundComponent,

  DepartmentDetailComponent,DepartmentOverviewComponent

,DepartmentContactComponent];

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

@Component({

  selector: 'app-test',

  template: `

    <p>

     <form>

     <input type="text" id="txtComment"/>

     <button type="submit" > Submit </button>

     </form>

    </p>

  `,

  styles: [

  ]

})

export class TestComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {

  }

}

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

@Component({

  selector: 'app-test',

  template: `

    <div>

     <form>

     <input type="text" id="txtComment"/>

     <button type="submit"> Submit </button>

     </form>

    </div>

  `,

  styles: [

  ]

})

export class TestComponent implements OnInit {

 name : any;

  constructor() { }

  ngOnInit(): void {

    this.name ="Hello";

  }

}

<div style="text-align: center;">

<h1> Routing & Navigtaion </h1>

</div>

dddd

<a routerLink="/departments" routerLinkActive="active"> Department List </a>

<a routerLink="/test" routerLinkActive="active"> Test </a>

<!--  -->

<a routerLink="/employees" routerLinkActive="active"> Employee List </a>

<router-outlet></router-outlet>

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

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

import { Observable } from 'rxjs';

import { TestComponent } from '../test/test.component';

@Injectable({

  providedIn: 'root'

})

export class CheckGuard implements CanDeactivate<TestComponent> {

  canDeactivate(

    test : TestComponent,

    route: ActivatedRouteSnapshot,

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

    if(test.name!="")

    {

      return window.confirm("There is some unsaved data. Do you really want to navigate ?")

    }

      return true;

  }

}