import { HttpErrorResponse, HttpEvent, HttpHandler, HttpInterceptor, HttpRequest } from "@angular/common/http";

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

import { Observable, throwError } from "rxjs";

import { UserAuthService } from "./user-auth.service";

import { catchError} from 'rxjs/operators';

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

@Injectable()

export class AuthInterceptor implements HttpInterceptor{

constructor(private userAuthService:UserAuthService,

private router:Router){}

//this will create a header with jwt token and send it to the baceknd

intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {

// throw new Error("Method not implemented.")

if(req.headers.get('No-Auth') === 'True'){

return next.handle(req.clone());

}

const token = this.userAuthService.getToken();//getToken is ins user.auth.ts which gets token from local storage and passs it to interceptor, if we do not enter anything it will send nothing

//req = this.addToken(req,token); this addToken function will add token value in the header even if it is nul hence we need to add a null condition as well.

if(token){

req = this.addToken(req,token);

}

//with thid if conditn it will addToken only when some token value is present.

console.log(req);// will print req in console

//through following code we are going to backend taking the token within

return next.handle(req).pipe(

catchError(

(err:HttpErrorResponse) => {

console.log(err.status);

if (err.status == 401) {

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

}else if(err.status == 403){

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

}

return throwError("Something went wrong");

}

)

);

}

private addToken(request:HttpRequest<any>, token:string){

return request.clone(

{

setHeaders:{

Authorization : `Bearer ${token}`

}

}

);

}

}

//check whether it contains header like NO\_AUTH --> we will not add jwtToken and it will return as it is