import { HttpRequest, HttpHandler, HttpEvent } from '@angular/common/http';

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

import { Observable, from, lastValueFrom, catchError, throwError } from 'rxjs';

import { TokenService } from '../token/token.service';

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

@Injectable({

providedIn: 'root'

})

export class AuthInterceptorService {

constructor(

private tokenServ: TokenService,

private auth: AuthService,

) { }

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

return from(this.handleReq(req, next));

}

private async handleReq(req: HttpRequest<any>, next: HttpHandler): Promise<HttpEvent<any>> {

const token: any = this.tokenServ.getToken();

if (!!token) {

req = req.clone({

setHeaders: {

Authorization: `${token}`

}

});

} else {

this.auth.logout();

}

return lastValueFrom(next.handle(req)

.pipe(

catchError(error => {

const messages = ['jwt expired', 'Session Expired'];

if ((error.status === 401) && messages.includes(error?.error.message)) {

this.auth.logout();

}

if (error.status === 403) {

console.warn('You can not access this resource');

}

return throwError(() => error);

})

));

}

}