import { HttpClient } from '@angular/common/http';

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

import { Observable } from 'rxjs';

import { Credentials } from '../credentials';

@Injectable({

providedIn: 'root'

})

export class LoginService {

constructor(private http: HttpClient) { }

baseUrl = 'http://localhost:8080';

//current user: user which is logged in

public getCurrentUser(): Observable<any> {

return this.http.get(`${this.baseUrl}/current-user`);

}

//generate token

public generateToken(credentials: Credentials): Observable<any> {

return this.http.post<Object>(`${this.baseUrl}/generate-token`, credentials);

}

//change password

public changePassword(credentials: Credentials): Observable<Object> {

return this.http.put<Object>(`${this.baseUrl}/change-password`, credentials);

}

//login user: set token in local storage

public loginUser(token: any) {

localStorage.setItem('token', token);

return true;

}

//is logged in: user is logged in or not

public isLoggedIn() {

let tokenStr = localStorage.getItem('token');

if (tokenStr == undefined || tokenStr == null || tokenStr == '') {

return false;

} else {

return true;

}

}

//logout user: remove token from local storage

public logout() {

localStorage.removeItem('token');

localStorage.removeItem('user');

return true;

}

//get token

public getToken() {

return localStorage.getItem('token');

}

//set user details in local storage

public setUserDetails(user: any) {

localStorage.setItem('user', JSON.stringify(user));

}

//get user details from local storage

public getUserDetails() {

let user = localStorage.getItem('user');

if (user != null) {

return JSON.parse(user);

} else {

this.logout();

return null;

}

}

//get user role

public getUserRole() {

let user = this.getUserDetails();

return user.authorities[0].authority;

}

}