import { AuthInterceptor } from '../shared/auth.interceptor';

import { LoggingInterceptor } from '../shared/logging.interceptor';

@NgModule({declarations: [

HeaderComponent,

HomeComponent

],imports: [

SharedModule,

AppRoutingModule

],exports: [

AppRoutingModule,

HeaderComponent

**],providers: [**

ShoppingListService,

RecipeService,

**DataStorageService,**

**AuthService**,

**{provide: HTTP\_INTERCEPTORS, useClass: AuthInterceptor, multi: true},**

**{provide: HTTP\_INTERCEPTORS, useClass: LoggingInterceptor, multi: true}**

]}) export class **CoreModule** {}------------------------------------------------------------------------

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

import { Observable } from 'rxjs/Observable';

import 'rxjs/add/operator/do';

export class **LoggingInterceptor** implements **HttpInterceptor** {

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

return **next.handle(req).do(**

**event** => {

console.log('Logging interceptor', event);

})}}

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

import { Observable } from 'rxjs/Observable';

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

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

**@Injectable()**

export **class AuthInterceptor** implements **HttpInterceptor** {

**constructor(private authService: AuthService)** {}

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

console.log('Intercepted!', **req**);

// const copiedReq = req.clone({**headers**: req.**headers**.set('', '')});

const copiedReq = **req.clone(**{**params**: req.**params**.set('auth', this.**authService**.**getToken**())});

return **next**.**handle**(**copiedReq**);

// return null;

}}

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

**import { HttpClient, HttpHeaders, HttpParams, HttpRequest } from '@angular/common/http';**

import 'rxjs/Rx';

**@Injectable()**

export class DataStorageService {

**constructor(private httpClient: HttpClient,**

**private recipeService: RecipeService,**

**private authService: AuthService) {**

**}**

storeRecipes() {

// **const headers = new HttpHeaders().set('Authorization', 'Bearer afdklasflaldf');**

// return this.**httpClient.put**('https://ng-recipe-book-3adbb.firebaseio.com/recipes.json', this.recipeService.getRecipes(), {

**// observe: 'body',**

**// params: new HttpParams().set('auth', token)**

// **// headers: headers**

// });

const req = new **HttpRequest**(**'PUT', 'https://ng-recipe-book-3adbb.firebaseio.com/recipes.json', this.recipeService.getRecipes(),** **{reportProgress: true});**

**return this.httpClient.request(req);**

}

**getRecipes() {**

// this.httpClient.get<Recipe[]>('https://ng-recipe-book-3adbb.firebaseio.com/recipes.json?auth=' + token)

**this.httpClient.get<Recipe[]>**('https://ng-recipe-book-3adbb.firebaseio.com/recipes.json', {

**observe: 'body',**

**responseType: 'json'**

})

**.map(**

(recipes) => {

console.log(recipes);

**for (let recipe of recipes) {**

if **(!recipe['ingredients']) {**

**recipe['ingredients'] = [];**

}

}

return recipes;

}

)

**.subscribe(**

(recipes: Recipe[]) => {

**this.recipeService.setRecipes(recipes);**

}

);

}

}