Creating JWT interceptor

we will be creating the interceptor. First, browse to the folder where the angular app was created and run the following command to achieve that:

$ ng generate service AuthInterceptor

Then, navigate to src/app and edit the file app.module.ts to look like the code below. I imported the HTTP Module for HTTP calls. I also made the interceptor that was just created to be a provider, so that it will have global access to all HTTP calls.

// the modules required by the app are imported here

import { BrowserModule } from '@angular/platform-browser'; // this ensures the application will run on the browser

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

import { AppRoutingModule } from './app-routing.module'; // ensures the application have routing capabilities

import { AppComponent } from './app.component'; // made present for bootstrapping application on the launch

import { HTTP\_INTERCEPTORS, HttpClientModule } from '@angular/common/http'; // enables the application to communicate with the backend services

import { AuthInterceptorService } from './auth-interceptor.service'; // this will allow the app to automatically attach authorization information to requests

import { HomeComponent } from './home/home.component'; // implements the home route

@NgModule({

declarations: [

AppComponent,

HomeComponent

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule

],

providers: [

{ provide: HTTP\_INTERCEPTORS, useClass: AuthInterceptorService, multi: true }

],

bootstrap: [AppComponent]

})

export class AppModule { }

Next, edit src/app/auth-interceptor.service.ts to look like the code shown below:

import { Injectable } from '@angular/core'; // imports the class that provides local storage for token

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

import { catchError, filter, take, switchMap } from "rxjs/operators";

import { Observable, throwError } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class AuthInterceptorService implements HttpInterceptor {

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

console.log("Interception In Progress"); // Interception Stage

const token: string = localStorage.getItem('token'); // This retrieves a token from local storage

req = req.clone({ headers: req.headers.set('Authorization', 'Bearer ' + token) });// This clones HttpRequest and Authorization header with Bearer token added

req = req.clone({ headers: req.headers.set('Content-Type', 'application/json') });

req = req.clone({ headers: req.headers.set('Accept', 'application/json') });

return next.handle(req)

.pipe(

catchError((error: HttpErrorResponse) => {

// Catching Error Stage

if (error && error.status === 401) {

console.log("ERROR 401 UNAUTHORIZED") // in case of an error response the error message is displayed

}

const err = error.error.message || error.statusText;

return throwError(error); // any further errors are returned to frontend

})

);

}

}

Interception stage code explanation

In authentication, the token obtained from the server is stored locally. It will then be retrieved from local storage and the header httpRequest req, a cloned and Authorisation, Bearer: token header is added into it. Then the token is sent in the httpRequest header.

Error stage code explanation

In case of an error response or 401, 402, etc. error status, the pipe will help catch the error. The user will not authenticate due to a bad request or Unauthorized Request. The error in the call is returned to the frontend in case of further error requests.

Resourses: <https://www.section.io/engineering-education/getting-started-with-jwt-using-angular8-and-nodejs/>