# Project

* Site-08-Api-Ng-interceptor-membercards-photogallery
  + Copied from: Site-07-Api-Ng-userentity-datetimeext-relationships-seeddata-automapper
* For the “MySocialConnect-SPA” issue npm install command to install all packages
  + ng serve : to run the spa

# New & Updates

|  |  |
| --- | --- |
| New | Updates |
| /core/models-interfaces/photo-dto.model.ts | /core/services/helper.service.ts |
| /core/models-interfaces/user-dto.model.ts | environment |
| /core/constants/app-constants.ts | /site/members/member-list |
| /core/services/member.service.ts | /site/members/member-card |
| /core/interceptors/jwt | app.module.ts |
|  | /site/members/member-detail |
|  |  |

# Clear Database

If you want to clear database then drop it and recreate it

dotnet ef database drop

dotnet ef database update

# core/models-interfaces

## photo-dto.model.ts

> ng g interface core/models-interfaces/photoDto --type=model

export interface PhotoDto {

    id: number;

    url: string;

    isMain: boolean;

    publicId: string;

}

## user-dto.model.ts

> ng g interface core/models-interfaces/userDto --type=model

import { PhotoDto } from "./photo-dto.model"

export interface UserDto {

    id: number;

    guId: string;

    userName: string;

    photoUrl: string;

    age: number;

    displayName: string;

    gender: string;

    introduction: string;

    lookingFor: string;

    interests: string;

    city: string;

    country: string;

    photos: PhotoDto[];

    lastActive: Date;

    createdOn: Date;

    updatedOn: Date;

}

# core/constants

## app-constants.ts

> ng g class core/constants/appConstants --skip-tests

export class AppConstants {

    public static readonly Bearer = 'Bearer ';

}

# environments

|  |  |
| --- | --- |
| environment.ts | environment.development.ts |
| export const environment = {    production: true,    title: "MySocialConnect",    displayConsoleLog: false,    usebaseUrlHttps: true,    webApiBaseUrlHttps: 'api/',    webApiBaseUrlHttp: 'api/'  }; | export const environment = {    production: false,    title: "MySocialConnect",    displayConsoleLog: true,    usebaseUrlHttps: true,    webApiBaseUrlHttps: 'https://localhost:5001/api/',    webApiBaseUrlHttp: 'http://localhost:5000/api/'  }; |

# core/interceptors

## jwt.interceptor.ts

> ng g interceptor core/interceptors/jwt --skip-tests

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

import {

  HttpRequest,

  HttpHandler,

  HttpEvent,

  HttpInterceptor

} from '@angular/common/http';

import { Observable, take } from 'rxjs';

import { AccountService } from '../services/account.service';

import { LoggedInUserDto } from '../models-interfaces/logged-in-user-dto.model';

import { AppConstants } from '../constants/app-constants';

@Injectable()

export class JwtInterceptor implements HttpInterceptor {

  constructor(private accountService: AccountService) {}

  intercept(request: HttpRequest<unknown>, next: HttpHandler): Observable<HttpEvent<unknown>> {

    let loggedInUser: LoggedInUserDto;

    //subscribe. only taking the first item. no need to unsubscribe here, subscription will complete

    this.accountService.currentLoggedInUser$.pipe(take(1)).subscribe({

      next: user => {

        if(user){

          request = request.clone({

            setHeaders: {

              Authorization: `${AppConstants.Bearer}${user.token}`

            }

          })

        }

      }

    });

    return next.handle(request);

  }

}

## app.module.ts

add the jwt interceptor to app.module.ts

  providers: [

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

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

  ],

# core/services

## helper.service.ts

since ap/ moved to the environment file, the properties needs to be adjusted here

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

import { environment } from 'src/environments/environment';

@Injectable({

  providedIn: 'root'

})

export class HelperService {

  private title: string = environment.title;

  private isProduction: boolean = environment.production;

  private isLogConsole: boolean = environment.displayConsoleLog;

  private baseUrlServer: string = environment.usebaseUrlHttps ? environment.webApiBaseUrlHttps : environment.webApiBaseUrlHttp;

  //private baseUrlApi: string = `${this.baseUrlServer}api`;

  constructor() {

    if(this.isLogConsole){

      console.log(`HelperService isProduction: ${this.isProduction}`);

      console.log(`HelperService baseUrlServer: ${this.baseUrlServer}`);

      //console.log(`HelperService baseUrlApi: ${this.baseUrlApi}`);

    }

  }

  public Title: string = this.title;

  public IsProduction: boolean = this.isProduction;

  public IsLogConsole: boolean = this.isLogConsole;

  public logIf(text: any){

    if(!this.isLogConsole) return;

    console.log(text);

  }

  public logIfFrom(msg: any, from: string){

    if(!this.isLogConsole) return;

    if(from) this.logIf(`\*\*\*${from}\*\*\*`)

    if(msg) this.logIf(msg);

  }

  public logIfError(error: any, from: string){

    if(!this.isLogConsole) return;

    this.logIf("\*\*\*displayError\*\*\*")

    if(from) this.logIf(from);

    if(error) this.logIf(error);

  }

  public replaceKeyValue(text: string, key: string, value: any, printtext: string = ""): string{

    let newText = text.replace(key, value.toString());

    if(printtext){

      this.logIf(`${printtext}: ${newText}`);

    }

    return newText;

  }

  public BaseUrlServer: string = this.baseUrlServer;

  //public BaseUrlApi: string = this.baseUrlApi;

  //keys

  public keyId = "[id]";

  public keyName = "[name]";

  public keyGuid = "[guid]";

  public keyPhotoId = "[photoId]";

  //users

  private urlUsers: string = `${this.baseUrlServer}users`;

  public urlUsersAll: string = `${this.urlUsers}/`;

  public urlUserUpdate: string = `${this.urlUsers}/`;

  public urlUserGetById: string = `${this.urlUsers}/${this.keyId}/`;

  public urlUserGetByName: string = `${this.urlUsers}/${this.keyName}/name`;

  public urlUserGetByGuid: string = `${this.urlUsers}/${this.keyGuid}/guid`;

  public urlUserPhotoAdd: string = `${this.urlUsers}/add/photo`;

  public urlUserPhotoSetMain: string = `${this.urlUsers}/set/photo/${this.keyPhotoId}/main`;

  public urlUserPhotoDelete: string = `${this.urlUsers}/delete/${this.keyPhotoId}/photo`;

  //account

  private urlAccount: string = `${this.baseUrlServer}account`;

  public urlAccountRegister: string = `${this.urlAccount}/register`;

  public urlAccountLogin: string = `${this.urlAccount}/login`;

  public urlAccountCheckUser: string = `${this.urlAccount}/checkUser/${this.keyName}`;

  //sampe controller urls for testing errors

  private urlSample = `${this.baseUrlServer}sample`;

  public urlSampleAuth: string = `${this.urlSample}/auth`;

  public urlSampleUserNotFound: string = `${this.urlSample}/usernotfound`;

  public urlSampleServerError: string = `${this.urlSample}/servererror`;

  public urlSampleBadRequest: string = `${this.urlSample}/badrequest`;

  public urlSampleLogin: string = `${this.urlSample}/samplelogin`;

}

## member.service.ts

> ng g s core/services/member --skip-tests

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

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

import { Observable, map, of } from 'rxjs';

import { LocalStorageService } from './local-storage.service';

import { HelperService } from './helper.service';

import { HttpClientService } from './http-client.service';

import { AppConstants } from '../constants/app-constants';

import { UserDto } from '../models-interfaces/user-dto.model';

@Injectable({

  providedIn: 'root'

})

export class MemberService {

  //to pass the auth token to the api, later will use interceptor

  private httpOptions;

  //state mangement. We dont want to pull the user info every time

  members: UserDto[] = [];

  constructor(private localStorageService: LocalStorageService,

            private helperService: HelperService,

            private httpClientService: HttpClientService) {

    //to pass the auth token to the api, later will user interceptor

    this.httpOptions = {

      headers: new HttpHeaders({

        Authorization: AppConstants.Bearer + this.localStorageService.getUserToken

      })

    };

  }

  /\*\*

   \* A GET users method

   \* @returns Observable userDto[]

   \*/

  getMembers(): Observable<UserDto[]>{

    /\*

    if(this.members.length > 0){

      return of(this.members);

    }

    \*/

    const url = this.helperService.urlUsersAll;

    this.helperService.logIfFrom(url, "memberService getMembers");

    //return this.httpClientService.get<UserDto[]>(url, this.httpOptions)

    return this.httpClientService.get<UserDto[]>(url, this.httpOptions) //jwt interceptor

    .pipe(

      map((members: UserDto[]) => {

        this.members = members;

        return members;

      })

    );

  }

  /\*\*

   \* A GET user by guid method

   \* @returns Observable userDto

   \*/

  getMemberByGuid(guid: string): Observable<UserDto>{

    /\*

    if(this.members.length > 0){

      const member = this.members.find(x => x.guId === guid);

      if(member){

        return of(member);

      }

    }

    \*/

    const url = this.helperService.replaceKeyValue(this.helperService.urlUserGetByGuid, this.helperService.keyGuid, guid, "");

    this.helperService.logIfFrom(url, "memberService getMemberByGuid");

    //return this.httpClientService.get<UserDto>(url, this.httpOptions)

    return this.httpClientService.get<UserDto>(url, this.httpOptions) //jwt interceptor

      .pipe(

        map((member: UserDto) => {

          return member;

        })

      );

  }

}

# /site/members/member-card

## member-card.compoenent.css

.card:hover img{

    transform: scale(1.2, 1.2);

    transition-duration: 500ms;

    transition-timing-function: ease-out;

    opacity: 0.7;

}

.card img{

    transform: scale(1.0, 1.0);

    transition-duration: 500ms;

    transition-timing-function: ease-out;

}

.card-img-wrapper{

    overflow: hidden; /\*the image sould not overflow out on hover\*/

    position: relative; /\*since the button will be absolutely positioned\*/

}

/\*the 3 button to show\*/

.card-img-wrapper:hover .member-icons{

    bottom: 0;

    opacity: 1;

}

/\*the 3 buttons\*/

.member-icons{

    position: absolute;

    bottom: -30%; /\*do not show the buttons till user is govering on them\*/

    left: 0;

    right: 0;

    margin-left: auto;

    margin-right: auto;

    opacity: 0;

}

/\*the 3 buttons\*/

.animate-icons{

    transition: all 0.3s ease-in-out;

}

## member-card.component.ts

It will receive the member as input from the member-list page

import { Component, Input, OnDestroy, OnInit } from '@angular/core';

import { UserDto } from 'src/app/core/models-interfaces/user-dto.model';

@Component({

  selector: 'app-member-card',

  templateUrl: './member-card.component.html',

  styleUrls: ['./member-card.component.css']

})

export class MemberCardComponent implements OnInit, OnDestroy {

  @Input() memberIn: UserDto | undefined; // = <UserDto>{};

  constructor(){}

  ngOnDestroy(): void {

  }

  ngOnInit(): void {

  }

}

## member-card.compoenent.html

<div class="card mb-4" \*ngIf="memberIn">

    <div class="card-img-wrapper">

        <img src="{{memberIn.photoUrl || './assets/user.png'}}" alt="{{memberIn.displayName}}" class="card-img-top">

        <!--buttons will be absolutley positioned-->

        <ul class="list-inline member-icons animate-icons text-center">

            <li class="list-inline-item">

                <button class="btn btn-primary" routerLink="/members/detail/{{memberIn.guId}}/{{memberIn.displayName}}"><i class="fa fa-regular fa-user"></i></button>

            </li>

            <li class="list-inline-item">

                <button class="btn btn-primary"><i class="fa fa-thumbs-up"></i></button>

            </li>

            <li class="list-inline-item">

                <button class="btn btn-primary"><i class="fa fa-envelope"></i></button>

            </li>

        </ul>

    </div>

    <div class="card-body p-1">

        <h6 class="card-title text-center mb-1">

            <i class="fa fa-user me-2"></i>

            {{memberIn.displayName}}

        </h6>

        <p class="card-text text-muted text-center">{{memberIn.city}}</p>

    </div>

</div>

# /site/members/member-list

## member-list.component.ts

import { Component, OnDestroy, OnInit } from '@angular/core';

import { Subscription } from 'rxjs';

import { UserDto } from 'src/app/core/models-interfaces/user-dto.model';

import { MemberService } from 'src/app/core/services/member.service';

@Component({

  selector: 'app-member-list',

  templateUrl: './member-list.component.html',

  styleUrls: ['./member-list.component.css']

})

export class MemberListComponent implements OnInit, OnDestroy {

  members: UserDto[] = [];

  membersSubscription!: Subscription;

  constructor(private memberService: MemberService){}

  ngOnDestroy(): void {

    if(this.membersSubscription) this.membersSubscription.unsubscribe();

  }

  ngOnInit(): void {

    this.loadMembers();

  }

  loadMembers(){

    this.membersSubscription = this.memberService.getMembers().subscribe({

      next: (members: UserDto[]) => {

        this.members = members;

      },

      error: e => {},

      complete: () => {}

    })

  }

}

## member-list.compoenent.html

use member-card and pass the member as an input

<div class="row">

    <div class="col-2" \*ngFor="let member of members">

        <app-member-card [memberIn]="member"></app-member-card>

    </div>

</div>

# member-detail.componenet

## tabs

Will be using tabsModule, so add it to the imports and exports array of the shared module

<https://valor-software.com/ngx-bootstrap/old/5.6.0/#/tabs>

### Tabs styles.css

Adding global styles for in the styles.css

/\*\*\*\*\*\*\*\*\* Members detail page tabs start \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

.tab-panel {

    border: 1px solid #ddd;

    padding: 10px;

    border-radius: 4px;

  }

  .nav-tabs > li.open, .member-tabset > .nav-tabs > li:hover {

    border-bottom: 4px solid #fbcdcf;

  }

  .member-tabset > .nav-tabs > li.open > a, .member-tabset > .nav-tabs > li:hover > a {

    border: 0;

    background: none !important;

    color: #333333;

  }

  .member-tabset > .nav-tabs > li.open > a > i, .member-tabset > .nav-tabs > li:hover > a > i {

    color: #a6a6a6;

  }

  .member-tabset > .nav-tabs > li.open .dropdown-menu, .member-tabset > .nav-tabs > li:hover .dropdown-menu {

    margin-top: 0px;

  }

  .member-tabset > .nav-tabs > li.active {

    border-bottom: 4px solid #E95420;

    position: relative;

  }

  .member-tabset > .nav-tabs > li.active > a {

    border: 0 !important;

    color: #333333;

  }

  .member-tabset > .nav-tabs > li.active > a > i {

    color: #404040;

  }

  .member-tabset > .tab-content {

    margin-top: -3px;

    background-color: #fff;

    border: 0;

    border-top: 1px solid #eee;

    padding: 15px 0;

  }

  /\*\*\*\*\*\*\*\*\* Members detail page tabs end \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## Photo Gallery

### Setup

* <https://www.npmjs.com/package/ng-gallery>
* <https://github.com/murhafsousli/ngx-gallery>
* Home page: <https://ngx-gallery.netlify.app/#/>
* Getting started: <https://ngx-gallery.netlify.app/#/getting-started/gallery>

>npm i ng-gallery @angular/cdk

If this gives an error then install a specific version

>npm i ng-gallery @angular/cdk@16.2.0

ng-gallery is stand alone component so it cannot be used inside a non stand alone compoenet.

### member.component.ts as StandAloneCompoenet

Convert it to stand alone component and import CommonModule, TabsModule and GalleryModule

import { CommonModule } from '@angular/common';

import { Component, OnDestroy, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { TabsModule } from 'ngx-bootstrap/tabs';

import { GalleryModule } from 'ng-gallery';

import { ToastrService } from 'ngx-toastr';

import { Subscription } from 'rxjs';

import { UserDto } from '../../../core/models-interfaces/user-dto.model';

import { MemberService } from '../../../core/services/member.service';

@Component({

  selector: 'app-member-detail',

  standalone: true,

  templateUrl: './member-detail.component.html',

  styleUrls: ['./member-detail.component.css'],

  imports: [CommonModule, TabsModule, GalleryModule]

})

Note: And remove it from declarations array of the shared module as well.

### PhotoGallery styles

Put in styles.css

  /\*\*\*\*\*\*\*\*\* photo gallery styles start \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

  .gallery{

    height: 600px;

  }

  /\*\*\*\*\*\*\*\*\* photo gallery styles end \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## member-detail.component.ts

import { CommonModule } from '@angular/common';

import { Component, OnDestroy, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { TabsModule } from 'ngx-bootstrap/tabs';

import { GalleryItem, GalleryModule, ImageItem } from 'ng-gallery';

import { ToastrService } from 'ngx-toastr';

import { Subscription } from 'rxjs';

import { UserDto } from '../../../core/models-interfaces/user-dto.model';

import { MemberService } from '../../../core/services/member.service';

@Component({

  selector: 'app-member-detail',

  standalone: true,

  templateUrl: './member-detail.component.html',

  styleUrls: ['./member-detail.component.css'],

  imports: [CommonModule, TabsModule, GalleryModule]

})

export class MemberDetailComponent implements OnInit, OnDestroy {

  member: UserDto | undefined;

  guidParam: string | null = "";

  nameParam: string | null = "";

  images: GalleryItem[] = [];

  memberSubscription!: Subscription;

  constructor(private memberService: MemberService,

              private activatedRoute: ActivatedRoute,

              private router: Router,

              private toastr: ToastrService){}

  ngOnDestroy(): void {

    if(this.memberSubscription) this.memberSubscription.unsubscribe();

  }

  ngOnInit(): void {

    this.guidParam = this.activatedRoute.snapshot.paramMap.get('guid');

    this.nameParam = this.activatedRoute.snapshot.paramMap.get("name");

    if(!this.guidParam)

      this.naviagteToMember("GUID not received on the page", "User Error");

    this.loadMember();

  }

  loadMember(){

    this.memberSubscription = this.memberService.getMemberByGuid(this.guidParam!).subscribe({

      next: (member: UserDto) => {

        if(this.nameParam?.toLowerCase() !== member.displayName.toLowerCase())

          this.naviagteToMember("Mismatched user", "User Error");

        this.member = member;

        this.getImages();

      },

      error: e => {},

      complete: () => {}

    })

  }

  getImages(){

    if(!this.member || !this.member.photos || this.member.photos.length <= 0) return;

    for(const photo of this.member.photos){

      this.images.push(new ImageItem({src: photo.url, thumb: photo.url} ))

    }

  }

  naviagteToMember(error: string, errorTitle: string){

    if(error)

      this.toastr.error(error, errorTitle);

    this.router.navigate(["/members"]);

  }

}

## member-detail.component.html

<div class="row" \*ngIf="member">

    <div class="col-4">

        <!--left start-->

        <div class="card">

            <img src="{{member.photoUrl || './assets/user.png'}}" alt="{{member.displayName | titlecase}}"

                class="card-img-top img-thumbnail">

            <div class="card-body">

                <div>

                    <strong>Location:</strong>

                    <p>{{member.city}}, {{member.country}}</p>

                </div>

                <div>

                    <strong>Age:</strong>

                    <p>{{member.age}}</p>

                </div>

                <div>

                    <strong>Last active:</strong>

                    <p>{{member.lastActive}}</p>

                </div>

                <div>

                    <strong>Member since:</strong>

                    <p>{{member.createdOn}}</p>

                </div>

            </div>

            <div class="card-footer">

                <div class="btn-group d-flex">

                    <button class="btn btn-primary">Like</button>

                    <button class="btn btn-success">Messages</button>

                </div>

            </div>

        </div>

         <!--left end-->

    </div>

    <div class="col-8">

        <!--right start-->

        <tabset class="member-tabset">

            <tab heading="About {{member.displayName | titlecase }}">

                <h4>Description</h4>

                <p>{{member.introduction}}</p>

                <h4>Looking for</h4>

                <p>{{member.lookingFor}}</p>

            </tab>

            <tab heading="Interests">

                <h4>Interests</h4>

                <p>{{member.interests}}</p>

            </tab>

            <tab heading="Photos" #photoTab="tab"> <!--a template reference variable-->

                <gallery \*ngIf="photoTab.active" [items]="images" class="gallery"></gallery>

            </tab>

            <tab heading="Messages">

                <p>Messages here</p>

            </tab>

        </tabset>

        <!--right end-->

    </div>

</div>