Table of Contents

[Current Resources with Changes 3](#_Toc114431107)

[New Resources Added 3](#_Toc114431108)

[Styles.css 3](#_Toc114431109)

[User.png 3](#_Toc114431110)

[Update Members Detail Route 3](#_Toc114431111)

[app-routing.module.ts 3](#_Toc114431112)

[Add AppRoutingModule, TabsModule to SharedModule 4](#_Toc114431113)

[/core/modules/shared.module.ts 4](#_Toc114431114)

[/app.module.ts 4](#_Toc114431115)

[Models 4](#_Toc114431116)

[core/models/photoDto.model.ts 4](#_Toc114431117)

[core/models/userDto.model.ts 4](#_Toc114431118)

[JWT Interceptor 4](#_Toc114431119)

[Traditional Token Handler 4](#_Toc114431120)

[JWT Interceptor 5](#_Toc114431121)

[Add to app.module.ts 5](#_Toc114431122)

[jwt.interceptor.ts 5](#_Toc114431123)

[Services 5](#_Toc114431124)

[/core/services/local-storage.service.ts 5](#_Toc114431125)

[/core/services/api-url.service.ts 5](#_Toc114431126)

[/core/services/http-client.service.ts 6](#_Toc114431127)

[/core/services/members.service.ts 6](#_Toc114431128)

[Photo Gallery 7](#_Toc114431129)

[Components 8](#_Toc114431130)

[/site/members/member-card 8](#_Toc114431131)

[Create New 8](#_Toc114431132)

[member-card.component.ts 8](#_Toc114431133)

[member-card.component.html 8](#_Toc114431134)

[member-card.component.css 8](#_Toc114431135)

[/site/members/member-detail 9](#_Toc114431136)

[member-detail.component.ts 9](#_Toc114431137)

[member-detail.component.html 11](#_Toc114431138)

[member-detail.component.css 12](#_Toc114431139)

[tab styles 12](#_Toc114431140)

[/site/members/member-list 13](#_Toc114431141)

[member-list.component.ts 13](#_Toc114431142)

[member-list.component.html 14](#_Toc114431143)

# Current Resources with Changes

1. /core/modules/shared.module.ts
2. /core/models/userDto.model.ts
3. /core/services/api-url.service.ts
4. /core/services/http-client.service.ts
5. /core/services/local-storage.serivce.ts
6. /site/members/member-list
7. app-routing.module.ts
8. /site/members/member-detail
9. styles.css

# New Resources Added

1. /core/models/photoDto.model.ts
2. /core/services/members.service.ts
3. /site/members/member-card
4. /core/interceptors/jwt.interceptor.ts
5. styles-member-tabs.css
6. Photo Gallery
7. User.png

# Styles.css

.main-container {

    margin-top: 60px;

    margin-bottom: 65px; /\*with footer\*/

}

footer {

    position: fixed;

    bottom: 0;

    width: 100%;

    height: 60px;

    border-top-width: 5px;

    border-top-style: solid;

    border-top-color: #FF7F50;

    display: block;

    padding: 0 10px;

    margin-top: 10px;

    text-align: center;

    background-color: #fff;

 }

# User.png

Move the user.png to the src/assets folder. This will be a placeholder image when the image are not available.

# Update Members Detail Route

## app-routing.module.ts

put the guid and name both to be passed to the members detail page

{ path: 'members/detail/:guid/:name', component: MemberDetailComponent },

# Add AppRoutingModule, TabsModule to SharedModule

## /core/modules/shared.module.ts

1. Add AppRoutingModule to imports and exports array
2. Also add TabsModule to the important and exports array

AppRoutingModule,

TabsModule.forRoot()

## /app.module.ts

Comment the AppRoutingModule in the imports array since it is coming from the shared module now.

The routes will not be recognized that are specified inside the components that are declared in shared module otherwise.

For the tabs see following

<https://valor-software.com/ngx-bootstrap/#/components/tabs?tab=overview>

# Models

## core/models/photoDto.model.ts

export class PhotoDto {

    constructor(public id: number, public url: string, public isMain: boolean){}

}

## core/models/userDto.model.ts

import { PhotoDto } from "./photoDto.model";

export class userDto {

    constructor(public id: number, public guId: string, public userName: string, public photoUrl: string, public age: number,

                public displayName: string, public gender: string, public introduction: string,

                public lookingFor: string, public interests: string, public city: string, public country: string,

                public photos: PhotoDto[],

                public lastActive: Date, public createdOn: Date, public updatedOn: Date) { }

}

# JWT Interceptor

## Traditional Token Handler

Declare a variable to create option

  private httpOptions;

Fill the header information using the constructor. Check [local storage](#_/core/services/local-storage.servic) for the new get properties.

this.httpOptions = {

      headers: new HttpHeaders({

        Authorization: 'Bearer ' + localStorageService.getLoggedinToken

      })

    };

And then pass the options to the call

return this.httpClientService

      .get<userDto>(url, this.httpOptions);

## JWT Interceptor

Create the interceptor with command

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

### Add to app.module.ts

Add following to the providers array

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

### jwt.interceptor.ts

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 { UserTokenDto } from '../models/userTokenDto.model';

@Injectable()

export class JwtInterceptor implements HttpInterceptor {

  //when we login the current user is setup via account service. Check the login method

  constructor(private accountService: AccountService) {}

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

    let currentUser: UserTokenDto;

    //subscribe. only taking the first item

    this.accountService.currentUser$.pipe(take(1)).subscribe(user => currentUser = user);

    if (currentUser!) {

      request = request.clone({

        setHeaders: {

          Authorization: `Bearer ${currentUser.token}`

        }

      });

    }

    return next.handle(request);

  }

}

# Services

## /core/services/local-storage.service.ts

Properties added to get the username and token from the user placed during login

  //public properties to get some common pieces

  getLoggedInUser: UserTokenDto = this.getItem(this.\_keyUser);

  getLoggedinToken: string = this.getLoggedInUser?.token;

  getloggedinUserName: string = this.getLoggedInUser?.userName;

## /core/services/api-url.service.ts

Add the following to the api url service

  //users controller urls

  private usersBaseUrl = `${this.apiBaseUrl}users/`;

  usersAll = `${this.usersBaseUrl}`;

  userByGuIdReplace = "[guid]";

  userByGuId = `${this.usersBaseUrl}${this.userByGuIdReplace}/guid`; //replace [guid] with the id of the user

  userByIdReplace = "[id]";

  userById = `${this.usersBaseUrl}${this.userByIdReplace}/id`; //replace [id] with the id of the user

  userByNameReplace = "[name]";

  userByName = `${this.usersBaseUrl}${this.userByNameReplace}/name`; //replace [name] with the userName of the user

  userByName = `${this.usersBaseUrl}[name]/name`; //replace [name] with the userName of the user

## /core/services/http-client.service.ts

The get has changed.

  get<TReturn>(url: string, params = {}): Observable<TReturn> {

    return this.httpClient

      .get<TReturn>(url, params )

      .pipe(retry(this.retries));

  }

## /core/services/members.service.ts

Create a new service with command

* ng g s /core/services/members --skip-tests

and then add the following to it.

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

import { ApiUrlService } from './api-url.service';

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

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

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

import { userDto } from '../models/userDto.model';

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

import { Observable } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class MembersService {

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

  private httpOptions;

  constructor(private apiUrlService: ApiUrlService, private httpClientService: HttpClientService, private localStorageService: LocalStorageService) {

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

    //not using this any more, using the interceptor

    this.httpOptions = {

      headers: new HttpHeaders({

        Authorization: 'Bearer ' + localStorageService.getLoggedinToken

      })

    };

  }

  /\*\*

   \* A GET method

   \* @returns returns Observable userDto[]

   \*/

  getMembers() : Observable<userDto[]> {

    const url = this.apiUrlService.usersAll;

    if (environment.displayConsoleLog) console.log(`Users allUrl: ${url}`);

    //users end point is protected by authentication so need to send the token as well

    //check jwt interceptor for details

    /\*

    return this.httpClientService

      .get<userDto[]>(url, this.httpOptions);

    \*/

      return this.httpClientService.get<userDto[]>(url);

    }

  /\*\*

   \* A GET method

   \* @param id id to fetch

   \* @returns returns Observable userDto

   \*/

   getMemberByGuId(guid: string): Observable<userDto>{

    let url = this.apiUrlService.userByGuId;

    //replace [guid] with the guid in the url

    url = url.replace(this.apiUrlService.userByGuIdReplace, guid.toString());

    if (environment.displayConsoleLog) console.log(`Users by id url: ${url}`)

    //users end point is protected by authentication so need to send the token as well

    //check jwt interceptor for details

    /\*

    return this.httpClientService

      .get<userDto>(url, this.httpOptions);

      \*/

      return this.httpClientService.get<userDto>(url);

  }

  /\*\*

   \* A GET method

   \* @param id id to fetch

   \* @returns returns Observable userDto

   \*/

  getMemberById(id: number): Observable<userDto>{

    let url = this.apiUrlService.userById;

    //replace [id] with the id in the url

    url = url.replace(this.apiUrlService.userByIdReplace, id.toString());

    if (environment.displayConsoleLog) console.log(`Users by id url: ${url}`);

    //users end point is protected by authentication so need to send the token as well

    //check jwt interceptor for details

    /\*

    return this.httpClientService

      .get<userDto>(url, this.httpOptions);

      \*/

      return this.httpClientService.get<userDto>(url);

  }

  /\*\*

   \* A GET method

   \* @param userName userName to fetch

   \* @returns returns Observable userDto

   \*/

  getMemberByUserName(userName: string): Observable<userDto>{

    let url = this.apiUrlService.userByName;

    //replace [name] with the userName in the url

    url = url.replace(this.apiUrlService.userByNameReplace, userName);

    if (environment.displayConsoleLog) console.log(`Users by userName url: ${url}`);

    //users end point is protected by authentication so need to send the token as well

    //check jwt interceptor for details

    /\*return this.httpClientService

      .get<userDto>(url, this.httpOptions);

    \*/

      return this.httpClientService.get<userDto>(url);

  }

}

# Photo Gallery

<https://www.npmjs.com/package/@kolkov/ngx-gallery>

Run the command to setup the ngx-gallery

* npm install @kolkov/ngx-gallery

Check the package.json

"@kolkov/ngx-gallery": "^2.0.1",

Then add NgxGalleryModule to both imports and exports arrays

NgxGalleryModule

# Components

## /site/members/member-card

### Create New

Create a new component and at the same add it to the shared.module.ts

* ng g c /site/members/member-card --module=/core/modules/shared.module.ts --skip-tests

### member-card.component.ts

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

import { userDto } from 'src/app/core/models/userDto.model';

@Component({

  selector: 'app-member-card',

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

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

})

export class MemberCardComponent implements OnInit {

  @Input() member: userDto = <userDto>{};

  constructor() { }

  ngOnInit(): void {

  }

}

### member-card.component.html

<div class="card mb-4">

    <div class="card-img-wrapper">

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

        <!--on hover these button will display only-->

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

            <li class="list-inline-item"><button class="btn btn-primary" routerLink='/members/detail/{{ member.guId }}/{{ member.userName }}'><i class="fa fa-light fa-user"></i></button></li>

            <li class="list-inline-item"><button class="btn btn-primary"><i class="fa fa-light fa-thumbs-up"></i></button></li>

            <li class="list-inline-item"><button class="btn btn-primary"><i class="fa fa-light 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 mr-2"></i>

            {{ member.displayName }}

        </h6>

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

    </div>

</div>

### member-card.component.css

.card:hover img{

    /\*will make the image bigger, looks like we are zooming in\*/

    transform: scale(1.2, 1.2);

    transition-duration: 500ms;

    transition-timing-function: ease-out;

    /\*so as to fade out slightly\*/

    opacity: 0.7;

}

.card image {

    /\*return the image back to its normal size\*/

    transform: scale(1.0, 1.0);

    transition-duration: 500ms;

    transition-timing-function: ease-out;

}

.card-img-wrapper {

    /\*keep the image inside the parent div\*/

    overflow: hidden;

    position: relative;

}

.member-icons{

    position: absolute;

    bottom: -30%;

    left: 0;

    right: 0;

    margin-right: auto;

    margin-left: auto;

    opacity: 0;

}

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

    bottom: 0;

    opacity: 1;

}

.animate {

    transition: all 0.3s ease-in-out;

}

## /site/members/member-detail

For the tabs see <https://valor-software.com/ngx-bootstrap/#/components/tabs?tab=overview>

### member-detail.component.ts

Will receive the id and name from the calling page

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

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

import { Subscription } from 'rxjs';

import { ToastrService } from 'ngx-toastr';

import {NgxGalleryOptions} from '@kolkov/ngx-gallery';

import {NgxGalleryImage} from '@kolkov/ngx-gallery';

import {NgxGalleryAnimation} from '@kolkov/ngx-gallery';

import { MembersService } from '../../../core/services/members.service';

import { userDto } from '../../../core/models/userDto.model';

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

@Component({

  selector: 'app-member-detail',

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

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

})

export class MemberDetailComponent implements OnInit, OnDestroy {

  member: userDto = <userDto>{};

  guid: string = "";

  name: string = "";

  //ngx gallery

  galleryOptions!: NgxGalleryOptions[];

  galleryImages!: NgxGalleryImage[];

  paramSubscription!: Subscription;

  membersSubscription!: Subscription;

  constructor(private memberService: MembersService, private route: ActivatedRoute, private router: Router, private toastrService: ToastrService) {}

  ngOnInit(): void {

    //gallery optioins for the images

    this.setGalleryOptions();

    //there are two ways to get the id and name from the route

    ////we can get the param using snapshot, but this usually breaks when the item is clicked from inside the page and the url is the same as already showing

    //#1

    //this.route.snapshot.paramMap.get('id');

    //#2 way, preferred

    this.paramSubscription = this.route.params.subscribe(params => {

      this.guid = params['guid'];

      this.name = params['name'];

      if (environment.displayConsoleLog) {

        console.log(`Param guid: ${this.guid} name: ${this.name}`);

      }

      if (!this.guid) {

        this.toastrService.error("GUID not received on the page", "User Error")

        this.router.navigate(['/members/list'])

      }

      //we have the id so get the data

      this.initData();

    });

  }

  ngOnDestroy(): void {

    if (this.paramSubscription) this.paramSubscription.unsubscribe();

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

  }

  //check https://www.npmjs.com/package/@kolkov/ngx-gallery

  setGalleryOptions() {

    this.galleryOptions = [

      {

        width: '500px',

        height: '500px',

        imagePercent: 100,

        thumbnailsColumns: 4,

        imageAnimation: NgxGalleryAnimation.Slide,

        preview: false

      }

    ]

  }

  //check https://www.npmjs.com/package/@kolkov/ngx-gallery

  getImages(): NgxGalleryImage[] {

    const imageUrls: NgxGalleryImage[] = [];

    if (!this.member || !this.member.photos) return imageUrls;

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

      imageUrls.push({

        small: photo?.url,

        medium: photo?.url,

        big: photo?.url

      })

    }

    if (environment.displayConsoleLog) console.log(imageUrls);

    return imageUrls;

  }

  initData() {

    this.membersSubscription = this.memberService.getMemberByGuId(this.guid).subscribe({

      next: (member: userDto) => {

        if (environment.displayConsoleLog) console.log(member);

        if (member.userName.toLowerCase() != this.name.toLowerCase()) {

          this.toastrService.error('Mismatched user');

          this.router.navigate(['/members/list']);

        }

        this.member = member;

        //image gallery

        this.galleryImages = this.getImages();

      }, error: e => {

      }, complete: () => {

      }

    });

  }

}

### member-detail.component.html

For the tabs see <https://valor-software.com/ngx-bootstrap/#/components/tabs?tab=overview>

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

    <!--left side quick info start-->

    <div class="col-4">

        <div class="card">

            <img src="{{ member.photoUrl || './assets/user.png' }}" alt="{{ member.displayName}}" 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="button-group d-flex">

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

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

                </div>

            </div>

        </div>

    </div>

    <!--left side quick info end-->

    <!--right side detail starts-->

    <div class="col-8">

        <!--

        ngx-bootstrap tabs

        https://valor-software.com/ngx-bootstrap/#/components/tabs?tab=overview

        styles for the member-tabset will go in the styles.css.

        These cannot be put in the local style sheet

        -->

        <tabset class="member-tabset">

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

                <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">

                <div \*ngIf="galleryImages">

                    <!--applying local style rather than via a styles.css-->

                    <ngx-gallery [options]="galleryOptions" [images]="galleryImages" style="display: inline-block; margin-bottom: 20px;"></ngx-gallery>

                </div>

            </tab>

            <tab heading="Messages">

                <h4>Messages</h4>

                <p>Messages will go here</p>

            </tab>

        </tabset>

    </div>

    <!--right side detail ends-->

</div>

### member-detail.component.css

.img-thumbnail{

    margin: 25px;

    width: 85%;

    height: 85%;

}

.card-body{

    padding: 0 25px;

}

.card-footer{

    padding: 10px 15px;

    background-color: #fff;

    border-top: none;

}

### tab styles

#### styles-member-tabs.css

Create a new css file in source. Following will go in it

.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;

  }

#### angular.json

Go to angular.json and add the new style sheet to the styles array. Add it after styles.css

"src/styles-member-tabs.css"

Make sure to stop and restart the angular app

## /site/members/member-list

### member-list.component.ts

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

import { userDto } from '../../../core/models/userDto.model';

import { MembersService } from '../../../core/services/members.service';

@Component({

  selector: 'app-member-list',

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

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

})

export class MemberListComponent implements OnInit {

  members: userDto[] = [];

  constructor(private membersService: MembersService) { }

  ngOnInit(): void {

    this.loadMembers();

  }

  loadMembers() {

    this.membersService.getMembers().subscribe({

      next: (members : userDto[]) => {

        this.members = members;

        console.log(members);

      }, error: e => {

        //error will get displayed by the error interceptor

      }, complete: () => {}

    });

  }

}

### member-list.component.html

<div class="row">

    <div \*ngFor="let member of members" class="col-sm-4 col-md-2 col-12">

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

    </div>

</div>