# Project

* Site-13-Api-Ng-like
  + Copied from: Site-12-Api-Ng-paging-sorting-filtering-actionfilters-timeago-iqueryable
* For the “MySocialConnect-SPA” issue npm install command to install all packages
  + ng serve : to run the spa

# New & Updates

|  |  |
| --- | --- |
| New | Updates |
| /core/enums/z-user-like-type.ts | /core/services/helper.service.ts |
| /core/models-interfaces/like-params-dto.models.ts | /core/services/member.service.ts |
|  | /site/members/member-card |
|  | /site/lists |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Clear Database

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

dotnet ef database drop

dotnet ef database update

New migrations added so check the [webapi doc](012%20Project13%20-WebApi-like.docx).

# /core/enums

## z-user-like-type.ts

export enum ZUserLikeType {

    none = 0,

    liked = 1,

    likedBy = 2

}

# /core/models-interfaces

## like-params-dto.models.ts

> ng g class core/models-interfaces/likeParamsDto --type=model

import { HttpParams } from "@angular/common/http";

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

import { ZUserLikeType } from "../enums/z-user-like-type";

export class LikeParamsDto {

    pageNumber: number = AppConstants.PageNumber;

    pageSize: number = AppConstants.PageSize;

    userLikeType: ZUserLikeType = ZUserLikeType.liked;

    constructor(){}

    //helper method to build search prams

    getPaginationSearchParams(): HttpParams{

        let params = new HttpParams();

        params = params.append('pageNumber', this.pageNumber.toString());

        params = params.append('pagesize', this.pageSize.toString());

        return params;

    }

    //helper method to build search params

    getLikeSearchParams(): HttpParams{

        let params = this.getPaginationSearchParams();

        params = params.append('userLikeType', ZUserLikeType[this.userLikeType]);

        return params;

    }

}

# /core/services

## helper.service.ts

Add the new likes urls

  //likes

  private urlLikes: string = `${this.baseUrlServer}likes`;

  public urlLikeAdd: string = `${this.urlLikes}/${this.keyId}`;

  public urlLikeGetUsers: string = `${this.urlLikes}/`;

## member.service.ts

Add the two new methods to add and get the likes

  addLike(id: number){

    var url = this.helperService.replaceKeyValue(this.helperService.urlLikeAdd, this.helperService.keyId, id);

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

    return this.httpClientService.post(url, {});

  }

  getLike(likeParams: LikeParamsDto){

    //add the pagination and filtering params

    const params = likeParams.getLikeSearchParams();

    const url = this.helperService.urlLikeGetUsers;

    this.helperService.logIfFrom(url, "memberService getLike")

    this.helperService.logIf(params);

    //make the call, the result is not full userDto, has some properties

    return this.getPaginatedResult<Partial<UserDto[]>>(url, params);

  }

# /site/members

## member-card.component

Add the like feature

### member-card.component.ts

  onAddLike(member: UserDto){

    this.addLikeSubscription = this.memberService.addLike(member.id).subscribe({

      next: () => {

        this.toastrService.success(`You have liked ${member.displayName}`);

      },

      error: e => {

        //no need to do something here since interceptor will display

      },

      complete: () => {

        //no need to do something here since interceptor will display

      }

    })

  }

### member-card.component.html

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

                <button class="btn btn-primary" (click)="onAddLike(memberIn)"><i class="fa fa-thumbs-up"></i></button>

            </li>

# /site/lists

Display the likes

## lists.component.ts

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

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

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

import { ZUserLikeType } from '../../core/enums/z-user-like-type';

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

import { LikeParamsDto } from '../../core/models-interfaces/like-params-dto.model';

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

import { Subscription } from 'rxjs';

import { PaginatedResult } from 'src/app/core/models-interfaces/pagination/paginated-result.model';

import { Pagination } from 'src/app/core/models-interfaces/pagination/pagination.model';

@Component({

  selector: 'app-lists',

  templateUrl: './lists.component.html',

  styleUrls: ['./lists.component.css']

})

export class ListsComponent implements OnInit, OnDestroy {

  pagination!: Pagination;

  members: Partial<UserDto[]> = []

  likeParams: LikeParamsDto;

  userTypeString: string;

  //to be used in the component

  zUserLiketype = ZUserLikeType;

  likesSubscription!: Subscription;

  constructor(private memberService: MemberService, private helperService: HelperService){

    this.likeParams = new LikeParamsDto();

    this.userTypeString = ZUserLikeType[this.likeParams.userLikeType];

    //this.helperService.logIfFrom(this.userTypeString, "ListsComponenet");

    //this.helperService.logIf(this.likeParams);

  }

  ngOnDestroy(): void {

    if(this.likesSubscription) this.likesSubscription.unsubscribe();

  }

  ngOnInit(): void {

    this.loadLikes();

  }

  loadLikes(){

    const userTypeString = ZUserLikeType[this.likeParams.userLikeType];

    //if the user has asked for a different type then reset the number

    if(this.userTypeString != userTypeString){

      this.userTypeString = userTypeString;

      this.likeParams.pageNumber = AppConstants.PageNumber;

    }

    this.likesSubscription = this.memberService.getLike(this.likeParams).subscribe({

      next: (response: PaginatedResult<Partial<UserDto[]>>) => {

        if(response.result)

          this.members = response.result

        if(response.pagination)

          this.pagination = response.pagination;

      },

      error: e => {},

      complete: () => {}

    });

  }

  onPageChanged(event: any){

    if(this.likeParams)

      this.likeParams.pageNumber = event.page;

    this.loadLikes();

  }

}

## lists.component.html

<div class="text-center mt-3">

    <h2>{{ userTypeString === zUserLiketype[zUserLiketype.liked] ? "Members I like" : "Members who like me" }}</h2>

</div>

<div class="container mt-3">

    <div class="d-flex">

        <div class="btn-group" name="userLikeType">

            <button type="button" class="btn btn-small"

                [ngClass]="likeParams.userLikeType == zUserLiketype.liked ? 'btn-success' : 'btn-info' "

                btnRadio="{{ zUserLiketype.liked }}"

                [(ngModel)]="likeParams.userLikeType"

                (click)="loadLikes()"><i class="fa fa-arrow-circle-o-right" \*ngIf="likeParams.userLikeType == zUserLiketype.liked">&nbsp;</i>Members I like</button>

            <button type="button" class="btn btn-small"

                [ngClass]="likeParams.userLikeType == zUserLiketype.likedBy ? 'btn-success' : 'btn-info' "

                btnRadio="{{ zUserLiketype.likedBy }}"

                [(ngModel)]="likeParams.userLikeType"

                (click)="loadLikes()"><i class="fa fa-arrow-circle-o-right" \*ngIf="likeParams.userLikeType == zUserLiketype.likedBy">&nbsp;</i>Members who like me</button>

        </div>

    </div>

    <div class="row mt-3" \*ngIf="members">

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

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

        </div>

    </div>

</div>

<div class="d-flex justify-content-center" \*ngIf="pagination">

    <pagination

        [boundaryLinks]="true"

        [totalItems]="pagination.totalItems"

        [itemsPerPage]="pagination.itemsPerPage"

        [(ngModel)]="pagination.currentPage"

        (pageChanged)="onPageChanged($event)"

        previousText="&lsaquo;"

        nextText="&rsaquo;"

        firstText="&laquo;"

        lastText="&raquo;"

        >

    </pagination>

</div>