# Project

* Site-19-Api-Ng-UnitOfWork-OptQueries-CofirmService-MemMessage
  + Copied from: Site-18-Api-Ng-responsive nav
* For the “MySocialConnect-SPA” issue npm install command to install all packages
  + ng serve : to run the spa

# New & Updates

|  |  |
| --- | --- |
| New | Updates |
| /core/services/confirm.service.ts | /core/modules/shared.module |
| /site/modals/confirm-model.componenet | /site/members/member-messages.component |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Clear Database

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

dotnet ef database drop

dotnet ef database update

# Confirm Dialogue

Provide a custom confirm dialogue rather then using browser provide confirm.

This was originally created under Site-9

## /site/modal/confirm-modal

> ng g c site/modals/confirmModal --skip-tests --module=core/modules/shared

### confirm-modal.component.ts

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

import { BsModalRef } from 'ngx-bootstrap/modal';

@Component({

  selector: 'app-confirm-modal',

  templateUrl: './confirm-modal.component.html',

  styleUrls: ['./confirm-modal.component.css']

})

export class ConfirmModalComponent implements OnInit {

  title = '';

  message = '';

  btnOkText = '';

  btnCancelText = '';

  result = false;

  constructor(public bsModalRef: BsModalRef){}

  ngOnInit(): void {

  }

  confirm(){

    this.result = true;

    this.bsModalRef.hide();

  }

  cancel(){

    this.bsModalRef.hide();

  }

}

### confirm-modal.component.html

<div class="modal-header">

    <h4 class="modal-title pull-left"> {{title}}</h4>

</div>

<div class="modal-body">

    <p>{{message}}</p>

</div>

<div class="modal-footer">

    <button class="btn btn-sm btn-success" (click)="confirm()">{{btnOkText}}</button>

    <button class="btn btn-sm btn-danger" (click)="cancel()">{{btnCancelText}}</button>

</div>

Initially done via site-9 Can DeActivateGuard => prevent-unsaved-chnages.guard.ts

## /core/services/confirm-modal.service.ts

> ng g c site/modals/confirmModal --skip-tests --module=core/modules/shared

Add the service to create a dialogue to be used by all other items.

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

import { BsModalRef, BsModalService } from 'ngx-bootstrap/modal';

import { ConfirmModalComponent } from '../../site/modals/confirm-modal/confirm-modal.component';

import { Observable, map } from 'rxjs';

@Injectable({

  providedIn: 'root'

})

export class ConfirmService {

  //modal service also used in site/admin/userManagement component

  bsModalRef: BsModalRef<ConfirmModalComponent> = new BsModalRef<ConfirmModalComponent>();

  constructor(private modalService: BsModalService) { }

  confirm() : Observable<boolean>{

    return this.confirmCustom('Confirmation', 'Are you sure you want to do this?', 'Ok','Cancel');

  }

  confirmCustom (title: string, message: string, btnOkText: string, btnCancelText: string) : Observable<boolean>{

    const config = {

      //same property names

      initialState: {

        title,

        message,

        btnOkText,

        btnCancelText

      }

    }

    this.bsModalRef = this.modalService.show(ConfirmModalComponent, config);

    return this.bsModalRef.onHidden!.pipe(

      map(() => {

        return this.bsModalRef!.content!.result;

      })

    );

  }

}

## /core/guards/prevent-unsaved-changes.guard.ts

import { CanDeactivateFn } from '@angular/router';

import { MemberEditComponent } from '../../site/members/member-edit/member-edit.component';

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

import { ConfirmService } from '../services/confirm.service';

//this will not prevent when user will type a different URL

//for different URL check member-edit.component.ts

//export const preventUnsavedChangesGuard: CanDeactivateFn<unknown> = (component, currentRoute, currentState, nextState) => {

export const preventUnsavedChangesGuard: CanDeactivateFn<MemberEditComponent> = (component) => {

  const confirmService = inject(ConfirmService);

  if(component.editForm?.dirty){

    //return confirm('Are you sure you want to continue? Any unsaved changes will be lost.');

    return confirmService.confirm();

  }

  return true;

};

# Scrolling for Member Messages

Originally created under Site-14

## /site/members/member.messages.component

### member-messages.component.html

        <!--for scrolling the messages set the

            template reference variable

            style it

            use angular function ScrollTop

         -->

        <ul #scrollMemberMessages

            style="overflow: scroll; height: 500px;"

            [scrollTop]="scrollMemberMessages.scrollHeight"

            \*ngIf="messagesIn.length > 0" class="chat">

### meber-messages.cmponent.css

.chat li{

    margin-bottom: 10px;

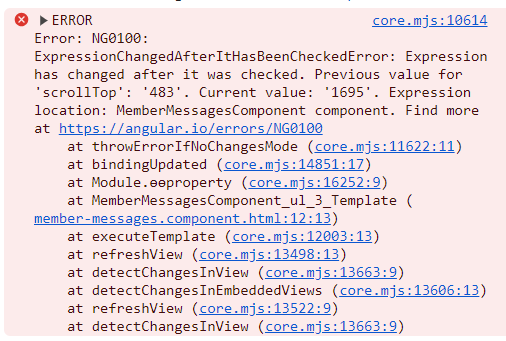
    padding-bottom: 10px;

    margin-right: 15px; /\*since scrolling now implemented\*/

}

### meber-messages.cmponent.ts

Above change will create a console error in development mode only and not production mode.



To fix it, change the “changeDetenction”

@Component({

  changeDetection: ChangeDetectionStrategy.OnPush, //due to scrolling the messages

  selector: 'app-member-messages',

  standalone: true,

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

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

  imports: [CommonModule, TimeagoModule, FormsModule]

})