Table of Contents

[Basics 2](#_Toc120029139)

[Current Resources with Changes 2](#_Toc120029140)

[New Resources Added 2](#_Toc120029141)

[/core/models 2](#_Toc120029142)

[photoDto.model.ts 2](#_Toc120029143)

[photoForApprovalDto.ts 2](#_Toc120029144)

[/core/services 2](#_Toc120029145)

[api-url.service.ts 2](#_Toc120029146)

[admin.service.ts 3](#_Toc120029147)

[Components 3](#_Toc120029148)

[/core/admin/photo-management 3](#_Toc120029149)

[photo-management.component.ts 3](#_Toc120029150)

[photo-management.component.html 4](#_Toc120029151)

[photo-management.component.css 4](#_Toc120029152)

[/site/members/photo-editor 4](#_Toc120029153)

[photo-editor.component.html 4](#_Toc120029154)

[photo-editor.component.css 4](#_Toc120029155)

# Basics

|  |  |
| --- | --- |
| WorkingFolder | Copy the content of “Site-07-UnitOfWork FinishingTouces” in “Site-08-PhotoManagement” and issue   * npm install * ng serve |

# Current Resources with Changes

1. /core/models/photoDto.model
2. /core/services/api-url.service
3. /site/admin/photo-management
4. /site/members/photo-editor

# New Resources Added

1. /core/models/photoForApprovalDto

Follow basics and then run “npm install” and “ng serve” commands.

# /core/models

## photoDto.model.ts

Add isApproved property

export class PhotoDto {

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

}

## photoForApprovalDto.ts

Create a new interface

export interface PhotoForApprovalDto{

  id: number;

  url: string;

  username: string;

  userId: number;

  userGuid: string;

  isApproved: boolean;

}

# /core/services

## api-url.service.ts

Add the new urls

  //admin controller urls

  private adminBaseUrl = `${this.apiBaseUrl}admin/`;

  adminRolesGuidReplace = "[guid]";

  adminPhotoIdReplace = "[photoId]";

  adminUsersWithroles = `${this.adminBaseUrl}users-with-roles`;

  adminEditRoles = `${this.adminBaseUrl}edit-roles/${this.adminRolesGuidReplace}`; //replace [guid] with user guids and pass in the new roles as params

  adminPhotosToModerate = `${this.adminBaseUrl}photos-to-moderate`;

  adminApprovePhoto = `${this.adminBaseUrl}approve-photo/${this.adminPhotoIdReplace}`; //replace [photoId] with photoId

  adminRejectPhoto = `${this.adminBaseUrl}reject-photo/${this.adminPhotoIdReplace}`; //replace [photoId] with photoId

## admin.service.ts

Add the getPhotosForApproval, approvePhoto and rejectPhoto methods

  getPhotosForApproval() {

    var url = this.apiUrlService.adminPhotosToModerate;

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

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

  }

  approvePhoto(photoId: number) {

    var url = this.apiUrlService.adminApprovePhoto.replace(this.apiUrlService.adminPhotoIdReplace, photoId.toString());

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

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

  }

  rejectPhoto(photoId: number) {

    var url = this.apiUrlService.adminRejectPhoto.replace(this.apiUrlService.adminPhotoIdReplace, photoId.toString());

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

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

  }

# Components

## /core/admin/photo-management

### photo-management.component.ts

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

import { PhotoForApprovalDto } from '../../../core/models/photoForApprovalDto';

import { AdminService } from '../../../core/services/admin.service';

@Component({

  selector: 'app-photo-management',

  templateUrl: './photo-management.component.html',

  styleUrls: ['./photo-management.component.css']

})

export class PhotoManagementComponent implements OnInit {

  photos!: PhotoForApprovalDto[];

  constructor(private adminService: AdminService) { }

  ngOnInit(): void {

    this.getPhotosForApproval();

  }

  getPhotosForApproval() {

    this.adminService.getPhotosForApproval().subscribe({

      next: (photos: PhotoForApprovalDto[]) => {

        this.photos = photos;

      },

      error: e => { },

      complete: () => { }

    });

  }

  approvePhoto(photoId: number) {

    this.adminService.approvePhoto(photoId).subscribe({

      next: () => {

        this.photos.slice(this.photos.findIndex(p => p.id == photoId), 1);

      }

    });

  }

  rejectPhoto(photoId: number) {

    this.adminService.rejectPhoto(photoId).subscribe({

      next: () => {

        this.photos.slice(this.photos.findIndex(p => p.id == photoId), 1);

      }

    });

  }

}

### photo-management.component.html

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

  <div class="col-sm-2" \*ngFor="let photo of photos">

    <h4>{{ photo.username }}</h4>

    <img src='{{photo.url}}' class='img-thumbnail p-1' alt='{{photo.username}}'>

    <div class='text-center'>

      <button class='btn btn-sm btn-success mr-1' (click)='approvePhoto(photo.id)'>Approve</button>

      <button class='btn btn-sm btndanger' (click)='rejectPhoto(photo.id)'>Reject</button>

    </div>

  </div>

</div>

### photo-management.component.css

img.img-thumbnail {

  height: 150px;

  min-width: 150px !important;

  margin-bottom: 2px;

}

## /site/members/photo-editor

* In the photo-editor when a user uploads photos ensure that some text is positioned absolutely on any currently unapproved photos as “awaiting approval” in red. Also reduce the opacity to 0.2 for any unapproved photos.
* Make sure that a user cannot select an unapproved photo as their main photo, but they can still delete unapproved photos.

### photo-editor.component.html

<div class="row">

    <!--display current user images-->

    <div class="col-2" \*ngFor="let photo of member.photos">

        <img src="{{ photo.url }}" alt="{{member.displayName}} photo URL {{ photo.url }}" class="img-thumbnail p-1"

          [class.not-approved]="!photo.isApproved"]>

        <div class="text-center img-text" \*ngIf="!photo.isApproved">

          <span class="text-danger">Awaiting approval</span>

        </div>

        <div class="text-center mt-1">

            <ng-container \*ngIf="photo.isMain">

                <button class="btn btn-sm btn-success active" [disabled]="photo.isMain || !photo.isApproved" title="Main Image"><i class="fa fa-star"></i></button>

            </ng-container>

            <ng-container \*ngIf="!photo.isMain">

                <button class="btn btn-sm btn-outline-success" title="Make Main" (click)="onSetMainPhoto(photo)"><i class="fa fa-check"></i></button>

            </ng-container>

            &nbsp;

            <button class="btn btn-sm btn-danger" title="Delete" [disabled]="photo.isMain" (click)="onDeletePhoto(photo)"><i class="fa fa-trash"></i></button>

        </div>

    </div>

</div>

### photo-editor.component.css

.not-approved {

  opacity: 0.2;

}

.img-wrapper {

  position: relative

}

.img-text {

  position: absolute;

  bottom: 30%;

}