package com.simplilearn.capstone2.controller;

import java.io.IOException;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.MediaType;

import org.springframework.http.ResponseEntity;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestPart;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.multipart.MultipartFile;

import com.simplilearn.capstone2.dao.MovieTicketDao;

import com.simplilearn.capstone2.entity.MovieTickets;

import com.simplilearn.capstone2.entity.Posters;

import com.simplilearn.capstone2.service.MovieTicketService;

@RestController

@CrossOrigin

public class MovieTicketController {

@Autowired

private MovieTicketService mtService;

// @Autowired

// private MovieTicketDao mtDao;

@PreAuthorize("hasRole('Admin')")

@PostMapping(value = { "/movieticket/add" }, consumes = { MediaType.MULTIPART\_FORM\_DATA\_VALUE })

public MovieTickets addNewTicket(@RequestPart("movieticket") MovieTickets mt,

@RequestPart("posters") MultipartFile[] file) {

// return mtService.addNewTicket(mt);

try {

Set<Posters> posters = uploadPoster(file);

mt.setMoviePoster(posters);

return mtService.addNewTicket(mt);

} catch (Exception e) {

System.out.println(e.getMessage());

return null;

}

}

// for processing these uploaded images

public Set<Posters> uploadPoster(MultipartFile[] multipartFiles) throws IOException {

Set<Posters> posters = new HashSet<>();

for (MultipartFile file : multipartFiles) {

Posters p = new Posters(file.getOriginalFilename(), file.getContentType(), file.getBytes());

posters.add(p);

}

return posters;

}

// @PreAuthorize("hasRole('Admin')")

@GetMapping({ "/movieticket/showall" })

public List<MovieTickets> getAllTickets() {

return mtService.getAllTickets();

}

@PreAuthorize("hasRole('Admin')")

@DeleteMapping({ "/deleteticket/{ticketId}" })

public void deleteTicket(@PathVariable("ticketId") Integer ticketId) {

mtService.deleteTicket(ticketId);

}

@GetMapping("getTicketById/{ticketId}")

public MovieTickets getTicketById(@PathVariable Integer ticketId) {

return mtService.getTicketsById(ticketId);

}

@PreAuthorize("hasRole('User')")

@GetMapping({ "/getTicketDetails/{isSingleTicketCheckout}/{ticketId}" })

public List<MovieTickets> getTicketDetails(@PathVariable(name = "isSingleTicketCheckout") boolean isSingleTicketCheckout,

@PathVariable(name = "ticketId")Integer ticketId) {

return mtService.getTicketDetails(isSingleTicketCheckout, ticketId);

}

}

/\*

\* For saving image file @RequestBody won't work,we need to send multipart file

\* multipart[] --> give access to store multiple images

\* (@RequestPart("movieticket") MovieTickets mt,

\*

\* @RequestPart("posters") MultipartFile[] file ) --> "moviticket" and "posters"

\* will be used at ui--> add-new-component.ts file isSingleTicketCheckout-->for

\* cart functionality

\*\*/