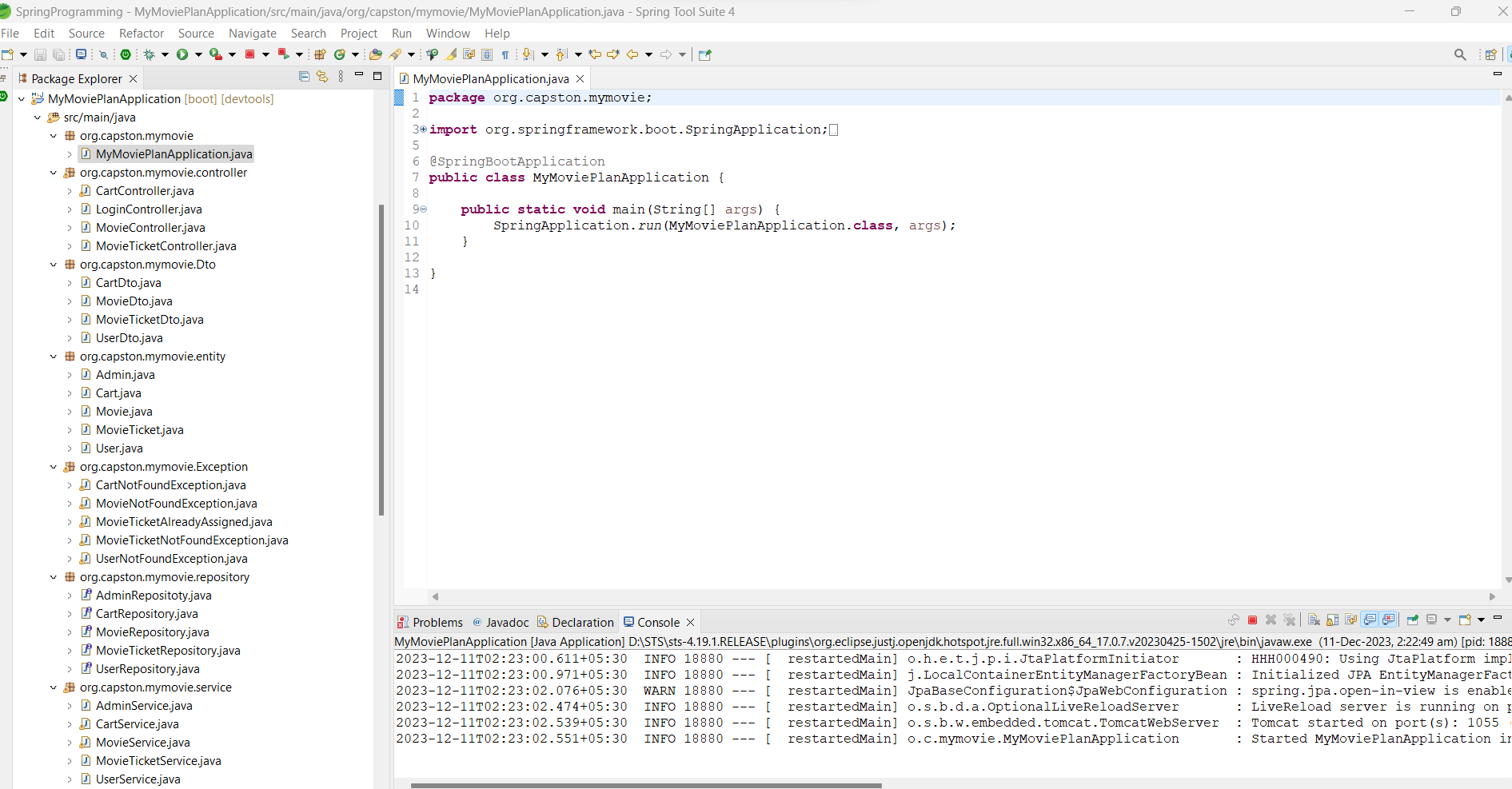
GitHub Link:



* Admin Entity:

package org.capston.mymovie.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class Admin {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private String email;

private String password;

public Admin() {

// TODO Auto-generated constructor stub

}

public Admin(Long id ,String email, String password) {

super();

this.id=id;

this.email = email;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

* User Entity:

package org.capston.mymovie.entity;

import java.util.ArrayList;

import java.util.List;

import org.capston.mymovie.Dto.UserDto;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.OneToOne;

@Entity

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private String fullname;

private String email;

private String password;

public User() {

// TODO Auto-generated constructor stub

}

@ManyToMany

@JsonIgnore

@JoinTable(name="user\_movieTicket",joinColumns=@JoinColumn(name="user\_id"),inverseJoinColumns = @JoinColumn(name="ticket\_id"))

private List<MovieTicket> movieTickets = new ArrayList<>();

public List<MovieTicket> getMovieTickets() {

return movieTickets;

}

public void setMovieTickets(List<MovieTicket> movieTickets) {

this.movieTickets = movieTickets;

}

@OneToOne(mappedBy ="user", cascade = CascadeType.ALL)

@JsonIgnore

private Cart cart;

public Cart getCart() {

return cart;

}

public void setCart(Cart cart) {

this.cart = cart;

}

public User(Long id, String fullname, String email, String password) {

super();

this.id = id;

this.fullname = fullname;

this.email = email;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getFullname() {

return fullname;

}

public void setFullname(String fullname) {

this.fullname = fullname;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public static User from(UserDto userDto) {

User user = new User();

user.setId(userDto.getId());

user.setFullname(userDto.getFullname());

user.setEmail(userDto.getEmail());

user.setPassword(userDto.getPassword());

user.setMovieTickets(userDto.getMovieTickets());

return user;

}

public void addmovieTikcet(MovieTicket movieTicket) {

movieTickets.add(movieTicket);

}

}

* Movie Entity:

package org.capston.mymovie.entity;

import java.util.ArrayList;

import java.util.List;

import org.capston.mymovie.Dto.MovieDto;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.OneToMany;

@Entity

public class Movie {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "movie\_id")

private Long id;

private String movie\_name;

private String movieImage;

private String category;

private String show\_time;

public Movie() {

}

public Movie( String movie\_name, String movieImage, String category, String show\_time,

List<MovieTicket> movieTicket) {

super();

this.movie\_name = movie\_name;

this.movieImage = movieImage;

this.category = category;

this.show\_time = show\_time;

this.movieTicket = movieTicket;

}

@OneToMany(mappedBy = "movie", cascade = CascadeType.ALL)

private List<MovieTicket> movieTicket = new ArrayList<>();

public String getMovieImage() {

return movieImage;

}

public void setMovieImage(String movieImage) {

this.movieImage = movieImage;

}

public Long getId() {

return id;

}

public List<MovieTicket> getMovieTicket() {

return movieTicket;

}

public void setMovieTicket(List<MovieTicket> movieTicket) {

this.movieTicket = movieTicket;

}

public void setId(Long id) {

this.id = id;

}

public String getMovie\_name() {

return movie\_name;

}

public void setMovie\_name(String movie\_name) {

this.movie\_name = movie\_name;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getShow\_time() {

return show\_time;

}

public void setShow\_time(String show\_time) {

this.show\_time = show\_time;

}

public void addMovieTicket(MovieTicket movieTicket) {

movieTicket.add(movieTicket);

}

public void removeMovieTicket(MovieTicket movieTicket) {

movieTicket.remove(movieTicket);

}

public static Movie from(MovieDto movieDto) {

Movie movie = new Movie();

movie.setId(movieDto.getId());

movie.setMovie\_name(movieDto.getMovie\_name());

movie.setMovieImage(movieDto.getMovieImage());

movie.setCategory(movieDto.getCategory());

movie.setShow\_time(movieDto.getShow\_time());

movie.setMovieTicket(movieDto.getMovieTickets());

return movie;

}

}

* MovieTicket Entity:

package org.capston.mymovie.entity;

import java.util.ArrayList;

import java.util.List;

import org.capston.mymovie.Dto.MovieTicketDto;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.ManyToOne;

@Entity

public class MovieTicket {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "ticket\_id")

private Long id;

private int total\_Ticket;

private Double price;

public MovieTicket() {

}

public MovieTicket(int total\_Ticket, Double price, Movie movie, List<User> user, List<Cart> cart) {

super();

this.total\_Ticket = total\_Ticket;

this.price = price;

this.movie = movie;

this.user = user;

//this.cart = cart;

}

@ManyToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY)

@JoinColumn(name = "movie\_id")

@JsonIgnore

private Movie movie;

@ManyToMany(mappedBy = "movieTickets", cascade = CascadeType.ALL)

@JsonIgnore

private List<User> user = new ArrayList<>();

@ManyToMany(mappedBy = "movieTickets", cascade = CascadeType.ALL)

@JsonIgnore

private List<Cart> cart = new ArrayList<>();

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public int getTotal\_Ticket() {

return total\_Ticket;

}

public void setTotal\_Ticket(int total\_Ticket) {

this.total\_Ticket = total\_Ticket;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public Movie getMovie() {

return movie;

}

public void setMovie(Movie movie) {

this.movie = movie;

}

public List<User> getUser() {

return user;

}

public void setUser(List<User> user) {

this.user = user;

}

public List<Cart> getCart() {

return cart;

}

public void setCart(List<Cart> cart) {

this.cart = cart;

}

public void add(MovieTicket movieTicket) {

}

public void remove(MovieTicket movieTicket) {

}

public void addCart(Cart cart) {

cart.add(cart);

}

// public void addUser(User user) {

// user.add(user);

//

// }

public static MovieTicket from(MovieTicketDto movieTicketDto) {

MovieTicket movieTicket = new MovieTicket();

movieTicket.setId(movieTicketDto.getId());

movieTicket.setTotal\_Ticket(movieTicketDto.getTotal\_Ticket());

movieTicket.setPrice(movieTicketDto.getPrice());

movieTicket.setMovie(movieTicketDto.getMovie());

return movieTicket;

}

}

* Cart Entity:

package org.capston.mymovie.entity;

import java.util.ArrayList;

import java.util.List;

import org.capston.mymovie.Dto.CartDto;

import com.fasterxml.jackson.annotation.JsonIgnore;

import jakarta.persistence.CascadeType;

import jakarta.persistence.Entity;

import jakarta.persistence.FetchType;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.JoinTable;

import jakarta.persistence.ManyToMany;

import jakarta.persistence.OneToOne;

@Entity

public class Cart {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Long id;

private Integer unit;

private Double totalPrice;

@ManyToMany

@JsonIgnore

@JoinTable(name = "cart\_MovieTicket", joinColumns = @JoinColumn(name = "cart\_id"), inverseJoinColumns = @JoinColumn(name = "ticket\_id"))

private List<MovieTicket> movieTickets = new ArrayList<>();

@OneToOne(cascade = CascadeType.ALL,fetch = FetchType.EAGER)

@JsonIgnore

@JoinColumn(name = "user\_id")

private User user;

public Cart() {

// TODO Auto-generated constructor stub

}

public Cart(Long id, Integer unit, Double totalPrice,List<MovieTicket> movieTickets, User user) {

super();

this.id = id;

this.unit = unit;

this.totalPrice = totalPrice;

this.movieTickets = movieTickets;

this.user = user;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Integer getUnit() {

return unit;

}

public void setUnit(Integer unit) {

this.unit = unit;

}

public Double getTotalPrice() {

return totalPrice;

}

public void setTotalPrice(Double totalPrice) {

this.totalPrice = totalPrice;

}

public List<MovieTicket> getMovieTickets() {

return movieTickets;

}

public void setMovieTickets(List<MovieTicket> movieTickets) {

this.movieTickets = movieTickets;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public static Cart from(CartDto cartDto) {

Cart cart = new Cart();

cart.setId(cartDto.getId());

cart.setUnit(cartDto.getUnit());

cart.setTotalPrice(cartDto.getTotalPrice());

cart.setMovieTickets(cartDto.getMovieTickets());

return cart;

}

public void addMovieTicket(MovieTicket movieTicket) {

movieTickets.add(movieTicket);

}

public void add(Cart cart) {

// TODO Auto-generated method stub

}

}

* CartDto:

package org.capston.mymovie.Dto;

import java.util.List;

import org.capston.mymovie.entity.Cart;

import org.capston.mymovie.entity.MovieTicket;

public class CartDto {

private Long id;

private Integer unit;

private Double totalPrice;

private List<MovieTicket> movieTickets;

public Integer getUnit() {

return unit;

}

public void setUnit(Integer unit) {

this.unit = unit;

}

public Double getTotalPrice() {

return totalPrice;

}

public void setTotalPrice(Double totalPrice) {

this.totalPrice = totalPrice;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public List<MovieTicket> getMovieTickets() {

return movieTickets;

}

public void setMovieTickets(List<MovieTicket> movieTickets) {

this.movieTickets = movieTickets;

}

public static CartDto from(Cart cart) {

CartDto cartDto = new CartDto();

cartDto.setId(cart.getId());

cartDto.setUnit(cart.getUnit());

cartDto.setTotalPrice(cart.getTotalPrice());

cartDto.setMovieTickets(cart.getMovieTickets());

return cartDto;

}

}

* MovieDto:

package org.capston.mymovie.Dto;

import java.util.List;

import org.capston.mymovie.entity.Movie;

import org.capston.mymovie.entity.MovieTicket;

public class MovieDto {

private Long id;

private String movie\_name;

private String movieImage;

private String category;

private String show\_time;

private List<MovieTicket> movieTickets;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getMovie\_name() {

return movie\_name;

}

public void setMovie\_name(String movie\_name) {

this.movie\_name = movie\_name;

}

public String getMovieImage() {

return movieImage;

}

public void setMovieImage(String movieImage) {

this.movieImage = movieImage;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getShow\_time() {

return show\_time;

}

public void setShow\_time(String show\_time) {

this.show\_time = show\_time;

}

public List<MovieTicket> getMovieTickets() {

return movieTickets;

}

public void setMovieTickets(List<MovieTicket> list) {

this.movieTickets = list;

}

public static MovieDto from(Movie movie) {

MovieDto movieDto = new MovieDto();

movieDto.setId(movie.getId());

movieDto.setMovie\_name(movie.getMovie\_name());

movieDto.setMovieImage(movie.getMovieImage());

movieDto.setCategory(movie.getCategory());

movieDto.setShow\_time(movie.getShow\_time());

movieDto.setMovieTickets(movie.getMovieTicket());

return movieDto;

}

}

* MovieTicketDto:

package org.capston.mymovie.Dto;

import org.capston.mymovie.entity.Movie;

import org.capston.mymovie.entity.MovieTicket;

public class MovieTicketDto {

private Long id;

private int total\_Ticket;

private Double price;

private Movie movie;

public Movie getMovie() {

return movie;

}

public void setMovie(Movie movie) {

this.movie = movie;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public int getTotal\_Ticket() {

return total\_Ticket;

}

public void setTotal\_Ticket(int total\_Ticket) {

this.total\_Ticket = total\_Ticket;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public static MovieTicketDto from(MovieTicket movieTicket) {

MovieTicketDto movieTicketDto = new MovieTicketDto();

movieTicketDto.setId(movieTicket.getId());

movieTicketDto.setTotal\_Ticket(movieTicket.getTotal\_Ticket());

movieTicketDto.setPrice(movieTicket.getPrice());

movieTicketDto.setMovie(movieTicket.getMovie());

return movieTicketDto;

}

}

CartService:

package org.capston.mymovie.service;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.StreamSupport;

import org.capston.mymovie.Exception.CartNotFoundException;

import org.capston.mymovie.entity.Cart;

import org.capston.mymovie.entity.MovieTicket;

import org.capston.mymovie.entity.User;

import org.capston.mymovie.repository.CartRepository;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

@Service

public class CartService {

@Autowired

private CartRepository cartRepository;

@Autowired

private MovieTicketService movieTicketService;

@Autowired

private UserService userService;

public Cart addToCart(Cart cart) {

return cartRepository.save(cart);

}

public Cart getCartById(Long id) {

return cartRepository.findById(id).orElseThrow(() -> new CartNotFoundException(id));

}

public List<Cart> getAllCartByUserId(Long id) {

return StreamSupport.stream(cartRepository.findAllByUserId(id).spliterator(), false).collect(Collectors.toList());

}

public List<Cart> getAllCarts() {

return StreamSupport.stream(cartRepository.findAll().spliterator(), false).collect(Collectors.toList());

}

@Transactional

public Cart updateCart(Cart cart, Long id) {

Cart updatecart = getCartById(id);

updatecart.setMovieTickets(cart.getMovieTickets());

return updatecart;

}

@Transactional

public Cart deleteCart(Long id) {

Cart cart = getCartById(id);

cartRepository.delete(cart);

return cart;

}

// @Transactional

// public Cart addMovieTicketToCart(Long cartId, Long movieTicketId) {

// Cart cart = getCartById(cartId);

// //User user=userService.getUserById(userId);

// MovieTicket movieTicket = movieTicketService.getMovieTicketById(movieTicketId);

// cart.addMovieTicket(movieTicket);

// movieTicket.addCart(cart);

// return cart;

// }

@Transactional

public Cart addMovieTicketToCart(Long cartId, Long userId) {

Cart cart = getCartById(cartId);

User user=userService.getUserById(userId);

// MovieTicket movieTicket = movieTicketService.getMovieTicketById(movieTicketId);

cart.setUser(user);

user.setCart(cart);

return cart;

}

}

* movieService:

package org.capston.mymovie.service;

import java.util.List;

import java.util.Objects;

import java.util.Optional;

import java.util.stream.Collectors;

import java.util.stream.StreamSupport;

import org.capston.mymovie.Exception.MovieNotFoundException;

import org.capston.mymovie.Exception.MovieTicketAlreadyAssigned;

import org.capston.mymovie.entity.Movie;

import org.capston.mymovie.entity.MovieTicket;

import org.capston.mymovie.repository.MovieRepository;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

@Service

public class MovieService {

@Autowired

private MovieRepository movieRepository;

@Autowired

private MovieTicketService movieTicketService;

public Movie addMovie(Movie movie) {

return movieRepository.save(movie);

}

public List<Movie> getAllMovie() {

// return movieRepository.findAll();

return StreamSupport.stream(movieRepository.findAll().spliterator(), false).collect(Collectors.toList());

}

public Movie getByIdMovie(Long id) {

// return movieRepository.findById(id).get();

return movieRepository.findById(id).orElseThrow(() -> new MovieNotFoundException(id));

}

@Transactional

public Movie updateMovie(Movie movie, Long id) {

Movie updateMovie = getByIdMovie(id);

updateMovie.setMovie\_name(movie.getMovie\_name());

updateMovie.setMovieImage(movie.getMovieImage());

updateMovie.setCategory(movie.getCategory());

updateMovie.setShow\_time(movie.getShow\_time());

updateMovie.setMovieTicket(movie.getMovieTicket());

return updateMovie;

}

public Movie deleteMovie(Long id) {

Movie movie = getByIdMovie(id);

movieRepository.delete(movie);

return movie;

}

@Transactional

public Movie addTicketToMovie(Long movieId, Long MovieTicketId) throws Exception {

Movie movie = getByIdMovie(movieId);

MovieTicket movieTicket = movieTicketService.getMovieTicketById(MovieTicketId);

if (Objects.nonNull(movieTicket.getMovie())) {

throw new MovieTicketAlreadyAssigned(MovieTicketId, movieId);

}

movie.addMovieTicket(movieTicket);

movieTicket.setMovie(movie);

return movie;

}

}

* MovieTicketService

package org.capston.mymovie.service;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.StreamSupport;

import org.capston.mymovie.Exception.MovieTicketNotFoundException;

import org.capston.mymovie.entity.MovieTicket;

import org.capston.mymovie.repository.MovieTicketRepository;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

@Service

public class MovieTicketService {

@Autowired

private MovieTicketRepository movieTicketRepository;

public MovieTicket addMovieTicket(MovieTicket movieTicket) {

return movieTicketRepository.save(movieTicket);

}

public MovieTicket getMovieTicketById(Long id) {

// return movieTicketRepository.findById(id).get();

return movieTicketRepository.findById(id).orElseThrow(() -> new MovieTicketNotFoundException(id));

}

public List<MovieTicket> getAllMovieTicketById(Long id)

{

return movieTicketRepository.findAllMovieTicketById(id);

}

public List<MovieTicket> getAllMovieTicket() {

// return movieTicketRepository.findAll();

return StreamSupport.stream(movieTicketRepository.findAll().spliterator(), false).collect(Collectors.toList());

}

@Transactional

public MovieTicket upadteMovieTicket(MovieTicket movieTicket, Long id) {

MovieTicket mv = getMovieTicketById(id);

mv.setTotal\_Ticket(movieTicket.getTotal\_Ticket());

mv.setPrice(movieTicket.getPrice());

mv.setMovie(movieTicket.getMovie());

return mv;

}

public MovieTicket deleteMovieTicket(Long id) {

MovieTicket movieTicket = getMovieTicketById(id);

movieTicketRepository.delete(movieTicket);

return movieTicket;

}

}

* UserService:

package org.capston.mymovie.service;

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.StreamSupport;

import org.capston.mymovie.Exception.MovieTicketNotFoundException;

import org.capston.mymovie.entity.MovieTicket;

import org.capston.mymovie.repository.MovieTicketRepository;

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

import org.springframework.stereotype.Service;

import jakarta.transaction.Transactional;

@Service

public class MovieTicketService {

@Autowired

private MovieTicketRepository movieTicketRepository;

public MovieTicket addMovieTicket(MovieTicket movieTicket) {

return movieTicketRepository.save(movieTicket);

}

public MovieTicket getMovieTicketById(Long id) {

// return movieTicketRepository.findById(id).get();

return movieTicketRepository.findById(id).orElseThrow(() -> new MovieTicketNotFoundException(id));

}

public List<MovieTicket> getAllMovieTicketById(Long id)

{

return movieTicketRepository.findAllMovieTicketById(id);

}

public List<MovieTicket> getAllMovieTicket() {

// return movieTicketRepository.findAll();

return StreamSupport.stream(movieTicketRepository.findAll().spliterator(), false).collect(Collectors.toList());

}

@Transactional

public MovieTicket upadteMovieTicket(MovieTicket movieTicket, Long id) {

MovieTicket mv = getMovieTicketById(id);

mv.setTotal\_Ticket(movieTicket.getTotal\_Ticket());

mv.setPrice(movieTicket.getPrice());

mv.setMovie(movieTicket.getMovie());

return mv;

}

public MovieTicket deleteMovieTicket(Long id) {

MovieTicket movieTicket = getMovieTicketById(id);

movieTicketRepository.delete(movieTicket);

return movieTicket;

}

}

* LoginController:

package org.capston.mymovie.controller;

import java.util.List;

import java.util.stream.Collectors;

import org.capston.mymovie.Dto.UserDto;

import org.capston.mymovie.entity.Admin;

import org.capston.mymovie.entity.User;

import org.capston.mymovie.service.AdminService;

import org.capston.mymovie.service.UserService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestMapping;

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

@RestController

@RequestMapping("/login")

@CrossOrigin("http://localhost:4200")

public class LoginController {

@Autowired

private AdminService adminService;

@Autowired

private UserService userService;

// \*\*\*\*\*\*\*\*\*\*\*\*ADMIN Operation\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@PostMapping("/add\_admin")

public Admin addAdmin(@RequestBody Admin admin) {

return adminService.addAdmin(admin);

}

@GetMapping("/getAll\_admin")

public List<Admin> getAllAdmin() {

return adminService.getAllAdmin();

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\* User Operation \*\*\*\*\*\*\*\*\*\*\*\*

@PostMapping("/add\_user")

public ResponseEntity<UserDto> addUser(@RequestBody UserDto userDto) {

User user = userService.addUser(User.from(userDto));

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@GetMapping("/getAll\_user")

public ResponseEntity<List<UserDto>> getAllUser() {

List<User> users = userService.getAllUser();

List<UserDto> userDtos = users.stream().map(UserDto::from).collect(Collectors.toList());

return new ResponseEntity<>(userDtos, HttpStatus.OK);

}

@GetMapping("/getuser/{id}")

public ResponseEntity<UserDto> getUserByid(@PathVariable Long id) {

User user = userService.getUserById(id);

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@PutMapping("/update\_user/{id}")

public ResponseEntity<UserDto> updateUser(@PathVariable Long id, @RequestBody UserDto userDto) {

User updateUser = userService.updateUser(id, User.from(userDto));

return new ResponseEntity<>(UserDto.from(updateUser), HttpStatus.OK);

}

@DeleteMapping("/delete\_user/{id}")

public ResponseEntity<UserDto> deleteUser(@PathVariable Long id) {

User user = userService.deleteUser(id);

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@PostMapping("/{userId}/movietickemovieTicketId}/add")

public ResponseEntity<UserDto> addUserToMovieTicket(@PathVariable Long userId,@PathVariable Long movieTicketId)

{

User user=userService.addTicketForUser(userId, movieTicketId);

return new ResponseEntity<>(UserDto.from(user),HttpStatus.OK);

}

}

* MovieController:

package org.capston.mymovie.controller;

import java.util.List;

import java.util.stream.Collectors;

import org.capston.mymovie.Dto.MovieDto;

import org.capston.mymovie.entity.Movie;

import org.capston.mymovie.service.MovieService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestMapping;

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

@RestController

@RequestMapping("/movie")

@CrossOrigin("http://localhost:4200")

public class MovieController {

@Autowired

private MovieService movieService;

@PostMapping("/add\_movie")

public ResponseEntity<MovieDto> addMovie(@RequestBody MovieDto movieDto) {

Movie movie = movieService.addMovie(Movie.from(movieDto));

return new ResponseEntity<>(MovieDto.from(movie), HttpStatus.OK);

}

@GetMapping("/getByid/{id}")

public ResponseEntity<MovieDto> getByID(@PathVariable Long id) {

Movie movie = movieService.getByIdMovie(id);

return new ResponseEntity<>(MovieDto.from(movie), HttpStatus.OK);

}

@GetMapping("/getAll\_movie")

public ResponseEntity<List<MovieDto>> getAllMovie() {

List<Movie> movies = movieService.getAllMovie();

List<MovieDto> movieDtos = movies.stream().map(MovieDto::from).collect(Collectors.toList());

return new ResponseEntity<>(movieDtos, HttpStatus.OK);

}

@PutMapping("/update\_movie/{id}")

public ResponseEntity<MovieDto> updateMovie(@PathVariable Long id, @RequestBody MovieDto movieDto) {

Movie editmovie = movieService.updateMovie(Movie.from(movieDto), id);

return new ResponseEntity<>(MovieDto.from(editmovie), HttpStatus.OK);

}

@DeleteMapping("/delete\_movie/{id}")

public ResponseEntity<MovieDto> deleteMovie(@PathVariable Long id) {

Movie movie = movieService.deleteMovie(id);

return new ResponseEntity<>(MovieDto.from(movie), HttpStatus.OK);

}

@PostMapping("{movieId}/movieTickets/{movieTicketId}/add")

public ResponseEntity<MovieDto> addTicketToMovie(@PathVariable Long movieId,@PathVariable Long movieTicketId) throws Exception

{

Movie movie=movieService.addTicketToMovie(movieId, movieTicketId);

return new ResponseEntity<>(MovieDto.from(movie),HttpStatus.OK);

}

}

* MovieTicketController:

package org.capston.mymovie.controller;

import java.util.List;

import java.util.stream.Collectors;

import org.capston.mymovie.Dto.MovieTicketDto;

import org.capston.mymovie.entity.MovieTicket;

import org.capston.mymovie.service.MovieService;

import org.capston.mymovie.service.MovieTicketService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestMapping;

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

@RestController

@RequestMapping("/ticket")

@CrossOrigin("http://localhost:4200")

public class MovieTicketController {

@Autowired

private MovieTicketService movieTicketService;

@Autowired

private MovieService movieService;

@PostMapping("/add\_ticket")

public ResponseEntity<MovieTicketDto> addMovieTicket(@RequestBody MovieTicketDto movieTicketDto) {

MovieTicket movieTicket = movieTicketService.addMovieTicket(MovieTicket.from(movieTicketDto));

return new ResponseEntity<>(MovieTicketDto.from(movieTicket), HttpStatus.OK);

}

@GetMapping("/get\_ticket/{id}")

public ResponseEntity<MovieTicketDto> getMovieTicketById(@PathVariable Long id) {

MovieTicket movieTicket = movieTicketService.getMovieTicketById(id);

movieService.getByIdMovie(id);

return new ResponseEntity<>(MovieTicketDto.from(movieTicket), HttpStatus.OK);

}

@GetMapping("/getAll\_ticket")

public ResponseEntity<List<MovieTicketDto>> getAllMovieTicket() {

List<MovieTicket> movieTickets = movieTicketService.getAllMovieTicket();

List<MovieTicketDto> movieTicketDtos = movieTickets.stream().map(MovieTicketDto::from).collect(Collectors.toList());

return new ResponseEntity<>(movieTicketDtos, HttpStatus.OK);

}

@PutMapping("/update\_ticket/{id}")

public ResponseEntity<MovieTicketDto> UpdateMovie(@PathVariable Long id,

@RequestBody MovieTicketDto movieTicketDto) {

MovieTicket movieTicket = movieTicketService.upadteMovieTicket(MovieTicket.from(movieTicketDto), id);

return new ResponseEntity<>(MovieTicketDto.from(movieTicket), HttpStatus.OK);

}

@DeleteMapping("/delete/{id}")

public ResponseEntity<MovieTicketDto> deleteMovie(@PathVariable Long id) {

MovieTicket movieTicket = movieTicketService.deleteMovieTicket(id);

return new ResponseEntity<>(MovieTicketDto.from(movieTicket), HttpStatus.OK);

}

}

* Login Controller:

package org.capston.mymovie.controller;

import java.util.List;

import java.util.stream.Collectors;

import org.capston.mymovie.Dto.UserDto;

import org.capston.mymovie.entity.Admin;

import org.capston.mymovie.entity.User;

import org.capston.mymovie.service.AdminService;

import org.capston.mymovie.service.UserService;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RequestMapping;

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

@RestController

@RequestMapping("/login")

@CrossOrigin("http://localhost:4200")

public class LoginController {

@Autowired

private AdminService adminService;

@Autowired

private UserService userService;

// \*\*\*\*\*\*\*\*\*\*\*\*ADMIN Operation\*\*\*\*\*\*\*\*\*\*\*\*\*\*

@PostMapping("/add\_admin")

public Admin addAdmin(@RequestBody Admin admin) {

return adminService.addAdmin(admin);

}

@GetMapping("/getAll\_admin")

public List<Admin> getAllAdmin() {

return adminService.getAllAdmin();

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\* User Operation \*\*\*\*\*\*\*\*\*\*\*\*

@PostMapping("/add\_user")

public ResponseEntity<UserDto> addUser(@RequestBody UserDto userDto) {

User user = userService.addUser(User.from(userDto));

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@GetMapping("/getAll\_user")

public ResponseEntity<List<UserDto>> getAllUser() {

List<User> users = userService.getAllUser();

List<UserDto> userDtos = users.stream().map(UserDto::from).collect(Collectors.toList());

return new ResponseEntity<>(userDtos, HttpStatus.OK);

}

@GetMapping("/getuser/{id}")

public ResponseEntity<UserDto> getUserByid(@PathVariable Long id) {

User user = userService.getUserById(id);

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@PutMapping("/update\_user/{id}")

public ResponseEntity<UserDto> updateUser(@PathVariable Long id, @RequestBody UserDto userDto) {

User updateUser = userService.updateUser(id, User.from(userDto));

return new ResponseEntity<>(UserDto.from(updateUser), HttpStatus.OK);

}

@DeleteMapping("/delete\_user/{id}")

public ResponseEntity<UserDto> deleteUser(@PathVariable Long id) {

User user = userService.deleteUser(id);

return new ResponseEntity<>(UserDto.from(user), HttpStatus.OK);

}

@PostMapping("/{userId}/movietickes/{movieTicketId}/add")

public ResponseEntity<UserDto> addUserToMovieTicket(@PathVariable Long userId,@PathVariable Long movieTicketId)

{

User user=userService.addTicketForUser(userId, movieTicketId);

return new ResponseEntity<>(UserDto.from(user),HttpStatus.OK);

}

}