Source code

# MySQL database connecting utility

spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/mymovieplan

spring.datasource.username=root

spring.datasource.password=123456789

# JPA property utility

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.show\_sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

package mypackage;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class MyMoviePlanBackendApplication {

public static void main(String[] args) {

SpringApplication.run(MyMoviePlanBackendApplication.class, args);

}

}

package mypackage;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

public class ServletInitializer extends SpringBootServletInitializer {

@Override

protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {

return application.sources(MyMoviePlanBackendApplication.class);

}

}

package mypackage.config;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Component;

@Component

public class SimpleCORSFilter implements Filter{

private final Logger log = LoggerFactory.getLogger(SimpleCORSFilter.class);

public SimpleCORSFilter() {

log.info("SimpleCORSFilter init");

}

@Override

public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)

throws IOException, ServletException {

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) res;

response.setHeader("Access-Control-Allow-Origin", request.getHeader("Origin"));

response.setHeader("Access-Control-Allow-Credentials", "true");

response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");

response.setHeader("Access-Control-Max-Age", "3600");

response.setHeader("Access-Control-Allow-Headers", "Content-Type, Accept, X-Requested-With, remember-me");

chain.doFilter(req, res);

}

@Override

public void init(FilterConfig filterConfig) {

}

@Override

public void destroy() {

}

}

package mypackage.controllers;

import java.util.List;

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

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.RequestBody;

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

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

import mypackage.models.Cart;

import mypackage.repositories.CartRepository;

@RestController

@RequestMapping("/api/v1/cart")

public class CartController {

@Autowired

CartRepository cartRepository;

@GetMapping("/movieTickets/all")

public List<Cart> getAllMovieTickets() {

return cartRepository.findAll();

}

@PostMapping("/movieTickets/add")

public Cart addMovieTicketToCart(@RequestBody(required = false) Cart cart) {

return cartRepository.save(cart);

}

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

public void deleteMovieTicketFromCart(@PathVariable Long id) {

cartRepository.deleteById(id);

}

@DeleteMapping("/movieTickets/delete/all")

public void deleteAllMovieTickets() {

cartRepository.deleteAll();

}

}

package mypackage.controllers;

import java.util.List;

import java.util.Optional;

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

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;

import mypackage.models.MovieTicket;

import mypackage.repositories.MovieTicketRepository;

@RestController

@RequestMapping("/api/v1/movieTicket")

public class MovieTicketController {

@Autowired

MovieTicketRepository movieTicketRepository;

@GetMapping("/movieTickets/all")

public List<MovieTicket> getAllMovieTickets() {

return movieTicketRepository.findAll();

}

@GetMapping("/movieTickets/getAParticularMovie/{movieName}")

public Optional<MovieTicket> getMovieTicketByMovieName(@PathVariable String movieName) {

return movieTicketRepository.getMovieTicketByMovieName(movieName);

}

@SuppressWarnings("deprecation")

@GetMapping("/movieTickets/getAParticularMovieById/{id}")

public MovieTicket getMovieTicketById(@PathVariable Long id) {

return movieTicketRepository.getById(id);

}

@PostMapping("/movieTickets/add")

public MovieTicket addMovieTicket(@RequestBody(required = false) MovieTicket movieTicket) {

return movieTicketRepository.save(movieTicket);

}

@PutMapping("/movieTickets/update")

public MovieTicket updateMovieTicket(@RequestBody(required = false) MovieTicket movieTicket) {

return movieTicketRepository.save(movieTicket);

}

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

public void deleteMovieTicket(@PathVariable Long id) {

movieTicketRepository.deleteById(id);

}

}

package mypackage.controllers;

import java.util.List;

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

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

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

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

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

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

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

import mypackage.models.MovieTicketSearchResults;

import mypackage.repositories.MovieTicketSearchResultsRepository;

@RestController

@RequestMapping("/api/v1/movieTicketSearchResults")

public class MovieTicketSearchResultsController {

@Autowired

private MovieTicketSearchResultsRepository movieTicketSearchResultsRepository;

@GetMapping("/movieTicket/search/results/all")

public List<MovieTicketSearchResults> getMovieTicketByMovieName() {

return movieTicketSearchResultsRepository.findAll();

}

@PostMapping("/movieTicket/search/results/add")

public MovieTicketSearchResults createMovieTicketSearchResult(

@RequestBody(required = false) String movieTicketSearchResults) {

MovieTicketSearchResults movieTicketSearchResults1 = new MovieTicketSearchResults();

movieTicketSearchResults1.setMovieName(movieTicketSearchResults);

return movieTicketSearchResultsRepository.save(movieTicketSearchResults1);

}

@DeleteMapping("/movieTicket/search/results/delete/")

public void deleteAllMovieTicketSearchResults() {

movieTicketSearchResultsRepository.deleteAll();

}

}

package mypackage.controllers;

import java.util.List;

import java.util.Optional;

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

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.RequestBody;

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

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

import mypackage.models.User;

import mypackage.repositories.UserRepository;

@RestController

@RequestMapping("/api/v1/user")

public class UserController {

@Autowired

UserRepository userRepository;

@PostMapping("/users/add")

public User addUser(@RequestBody(required = false) User user) {

return userRepository.save(user);

}

@GetMapping("/users/getParticularUser/{id}")

public Optional<User> getUserById(@PathVariable Long id) {

return userRepository.findById(id);

}

@GetMapping("/users/getAParticularUser/details")

public Optional<User> getUserDetails(@RequestBody(required = false) User user) {

return userRepository.getUserDetails(user.getId(), user.getFirstName(), user.getLastName(),

user.getUsername(), user.getEmail());

}

@GetMapping("/users/all")

public List<User> getAllUsers() {

return userRepository.findAll();

}

}

package mypackage.models;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "cart")

public class Cart {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private Long id;

@Column(name = "movie\_name")

private String movieName;

@Column(name = "show\_date")

private String showDate;

@Column(name = "show\_time")

private String showTime;

@Column(name = "showing\_location")

private String showingLocation;

@Column(name = "price")

private Long price;

public Cart() {}

public Cart(String movieName, String showDate, String showTime, String showingLocation, Long price) {

super();

this.movieName = movieName;

this.showDate = showDate;

this.showTime = showTime;

this.showingLocation = showingLocation;

this.price = price;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getShowDate() {

return showDate;

}

public void setShowDate(String showDate) {

this.showDate = showDate;

}

public String getShowTime() {

return showTime;

}

public void setShowTime(String showTime) {

this.showTime = showTime;

}

public String getShowingLocation() {

return showingLocation;

}

public void setShowingLocation(String showingLocation) {

this.showingLocation = showingLocation;

}

public Long getPrice() {

return price;

}

public void setPrice(Long price) {

this.price = price;

}

}

package mypackage.models;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "movie\_tickets")

public class MovieTicket {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private Long id;

@Column(name = "movie\_name")

private String movieName;

@Column(name = "show\_date")

private String showDate;

@Column(name = "show\_time")

private String showTime;

@Column(name = "showing\_location")

private String showingLocation;

@Column(name = "price")

private Long price;

public MovieTicket() {}

public MovieTicket(String movieName, String showDate, String showTime, String showingLocation,

Long price) {

super();

this.movieName = movieName;

this.showDate = showDate;

this.showTime = showTime;

this.showingLocation = showingLocation;

this.price = price;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

public String getShowDate() {

return showDate;

}

public void setShowDate(String showDate) {

this.showDate = showDate;

}

public String getShowTime() {

return showTime;

}

public void setShowTime(String showTime) {

this.showTime = showTime;

}

public String getShowingLocation() {

return showingLocation;

}

public void setShowingLocation(String showingLocation) {

this.showingLocation = showingLocation;

}

public Long getPrice() {

return price;

}

public void setPrice(Long price) {

this.price = price;

}

}

package mypackage.models;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "movie\_ticket\_search\_results")

public class MovieTicketSearchResults {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private long id;

public long getId() {

return id;

}

public void setId(long id) {

this.id = id;

}

@Column(name = "movie\_name")

private String movieName;

public MovieTicketSearchResults() { }

public MovieTicketSearchResults(String movieName) {

super();

this.movieName = movieName;

}

public String getMovieName() {

return movieName;

}

public void setMovieName(String movieName) {

this.movieName = movieName;

}

}

package mypackage.models;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "users")

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id")

private Long id;

@Column(name = "first\_name")

private String firstName;

@Column(name = "last\_name")

private String lastName;

@Column(name = "username")

private String username;

@Column(name = "email")

private String email;

@Column(name = "password")

private String password;

public User() {}

public User(String firstName, String lastName, String username, String email, String password) {

super();

this.firstName = firstName;

this.lastName = lastName;

this.username = username;

this.email = email;

this.password = password;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.email = lastName;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

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;

}

}

package mypackage.repositories;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import mypackage.models.Cart;

@Repository

public interface CartRepository extends JpaRepository<Cart, Long> {

}

package mypackage.repositories;

import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import mypackage.models.MovieTicket;

@Repository

public interface MovieTicketRepository extends JpaRepository<MovieTicket, Long> {

@Query(value = "SELECT \* FROM movie\_tickets WHERE movie\_name = :movieName", nativeQuery = true)

public Optional<MovieTicket> getMovieTicketByMovieName(@Param("movieName") String movieName);

}

package mypackage.repositories;

import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import mypackage.models.MovieTicketSearchResults;

@Repository

public interface MovieTicketSearchResultsRepository extends JpaRepository<MovieTicketSearchResults, Long> {

@Query(value = "SELECT \* FROM movie\_ticket\_search\_results WHERE movieName = :movieName", nativeQuery = true)

public Optional<MovieTicketSearchResults> getMovieTicketByMovieName(@Param("movieName") String movieName);

}

package mypackage.repositories;

import java.util.Optional;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.query.Param;

import org.springframework.stereotype.Repository;

import mypackage.models.User;

@Repository

public interface UserRepository extends JpaRepository<User, Long> {

@Query(value = "SELECT \* FROM users WHERE id = :id AND first\_name = :firstName AND "

+ "last\_name = :lastName AND username = :username AND email = :email", nativeQuery = true)

public Optional<User> getUserDetails(@Param("id") Long id,

@Param("firstName") String firstName,

@Param("lastName") String lastName,

@Param("username") String username,

@Param("email") String email);

}