package com.simplilearn.capstone2.service;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Collectors;

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

import org.springframework.stereotype.Service;

import com.simplilearn.capstone2.configuration.JwtRequestFilter;

import com.simplilearn.capstone2.dao.CartDao;

import com.simplilearn.capstone2.dao.MovieTicketDao;

import com.simplilearn.capstone2.dao.UserDao;

import com.simplilearn.capstone2.entity.Cart;

import com.simplilearn.capstone2.entity.MovieTickets;

import com.simplilearn.capstone2.entity.User;

@Service

public class MovieTicketService {

@Autowired //(required=true)

private MovieTicketDao mtDao;

@Autowired

private UserDao userDao;

@Autowired

private CartDao cartDao;

public MovieTickets addNewTicket(MovieTickets mt) {

return mtDao.save(mt); // this saves value in db and return the value saved in db

}

public List<MovieTickets> getAllTickets(){

return (List<MovieTickets>)mtDao.findAll();

}

public void deleteTicket(Integer ticketId) {

mtDao.deleteById(ticketId);

}

public MovieTickets getTicketsById(Integer ticketId) {

return mtDao.findById(ticketId).get();

}

public List<MovieTickets> getTicketDetails(boolean isSingleTicketCheckout,Integer ticketId) {

if(isSingleTicketCheckout && ticketId != 0) {

//we are checking out only one ticket

List<MovieTickets> list = new ArrayList<>();

MovieTickets ticket = mtDao.findById(ticketId).get(); //findById returns optional so we have to fetch product from so we user .get()

list.add(ticket);

return list;

}else {

//buying all tickets added to the cart.

String username = JwtRequestFilter.CURRENT\_USER; //we have usrname but we need all user details

User user = userDao.findById(username).get();

List<Cart> cart = cartDao.findByUser(user);

return cart.stream().map(x -> x.getMovieTicket()).collect(Collectors.toList());

}

}

}