Booking and Payment module

package com.booking;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

import org.springframework.cloud.openfeign.EnableFeignClients;

@SpringBootApplication

@EnableDiscoveryClient

@EnableFeignClients(basePackages = "com.booking.client")

public class BookingApplication {

public static void main(String[] args) {

SpringApplication.run(BookingApplication.class, args);

}

}

**package** com.booking.client;

**import** com.booking.DTO.TravelPackageDTO;

**import** com.booking.response.ApiResponse;

**import** org.springframework.cloud.openfeign.FeignClient;

**import** org.springframework.web.bind.annotation.\*;

@FeignClient(name = "travel-package-management", url = "http://localhost:8089/api/packages")

**public** **interface** TravelPackageClient {

@GetMapping("/{id}")

ApiResponse getPackageById(@PathVariable("id") Long id);

@GetMapping("/search/title/{title}")

ApiResponse searchByTitle(@PathVariable("title") String title);

@GetMapping("/search/price/{maxPrice}")

ApiResponse searchByPrice(@PathVariable("maxPrice") **double** maxPrice);

@GetMapping("/search/offer/{couponCode}")

ApiResponse searchByOffer(@PathVariable("couponCode") String couponCode);

}

**package** com.booking.controller;

**import** com.booking.entity.Booking;

**import** com.booking.DTO.TravelPackageDTO;

**import** com.booking.service.BookingService;

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

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.web.bind.annotation.\*;

**import** java.util.List;

@RestController

@RequestMapping("/api/bookings")

**public** **class** BookingController {

@Autowired

**private** BookingService bookingService;

@PostMapping

**public** Booking create(@RequestBody Booking booking) {

**return** bookingService.createBooking(booking);

}

@GetMapping

**public** List<Booking> getAll() {

**return** bookingService.getAllBookings();

}

@GetMapping("/{id}")

**public** Booking getById(@PathVariable Long id) {

**return** bookingService.getBookingById(id);

}

@DeleteMapping("/{id}")

**public** **void** delete(@PathVariable Long id) {

bookingService.deleteBooking(id);

}

@PutMapping("/cancel/{id}")

**public** ResponseEntity<String> cancelBooking(@PathVariable Long id) {

**return** bookingService.cancelBooking(id);

}

}

package com.booking.controller;

import com.booking.DTO.PaymentResponseDTO;

import com.booking.entity.Payment;

import com.booking.service.PaymentService;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/payments")

public class PaymentController {

@Autowired private PaymentService paymentService;

@PostMapping public PaymentResponseDTO process(@RequestBody Payment payment) {

Payment savedPayment = paymentService.processPayment(payment);

PaymentResponseDTO response = new PaymentResponseDTO();

response.setPaymentId(savedPayment.getPaymentId());

response.setUserId(savedPayment.getUserId());

response.setBookingId(savedPayment.getBookingId());

response.setAmount(savedPayment.getAmount());

response.setStatus(savedPayment.getStatus());

response.setPaymentMethod(savedPayment.getPaymentMethod());

response.setCurrency(savedPayment.getCurrency());

return response;

}

@GetMapping public List<Payment> getAll() { return paymentService.getAllPayments(); }

@GetMapping("/{id}") public Payment getById(@PathVariable Long id) { return paymentService.getPaymentById(id); }

}**package** com.booking.DTO;

**import** lombok.Data;

@Data

**public** **class** OfferDTO {

**private** String couponCode;

**private** String description;

**private** **int** discountPercentage;

**private** **boolean** active;

}

**package** com.booking.DTO;

**public** **class** PaymentResponseDTO {

**private** Long paymentId;

**private** Long userId;

**private** String bookingId;

**private** **double** amount;

**private** String status;

**private** String paymentMethod;

**private** String currency;

// Getters and Setters

**public** Long getPaymentId() {

**return** paymentId;

}

**public** **void** setPaymentId(Long paymentId) {

**this**.paymentId = paymentId;

}

**public** Long getUserId() {

**return** userId;

}

**public** **void** setUserId(Long userId) {

**this**.userId = userId;

}

**public** String getBookingId() {

**return** bookingId;

}

**public** **void** setBookingId(String bookingId) {

**this**.bookingId = bookingId;

}

**public** **double** getAmount() {

**return** amount;

}

**public** **void** setAmount(**double** amount) {

**this**.amount = amount;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String getPaymentMethod() {

**return** paymentMethod;

}

**public** **void** setPaymentMethod(String paymentMethod) {

**this**.paymentMethod = paymentMethod;

}

**public** String getCurrency() {

**return** currency;

}

**public** **void** setCurrency(String currency) {

**this**.currency = currency;

}

}

package com.booking.DTO;

import lombok.Data;

import java.time.LocalDate;

import java.util.List;

@Data

public class TravelPackageDTO {

private Long packageId;

private Long agentId;

private String title;

private String description;

private int duration;

private double price;

private int maxCapacity;

private LocalDate tripStartDate;

private LocalDate tripEndDate;

private List<String> highlights;

private OfferDTO offer;

}package com.booking.entity;

import java.time.LocalDate;

import jakarta.persistence.\*;

@Entity

public class Booking {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long bookingId;

private Long userId;

private String packageId;

private LocalDate startDate;

private LocalDate endDate;

private String status;

private Long paymentId;

// Getters and Setters

public Long getBookingId() {

return bookingId;

}

public void setBookingId(Long bookingId) {

this.bookingId = bookingId;

}

public Long getUserId() {

return userId;

}

public void setUserId(Long userId) {

this.userId = userId;

}

public String getPackageId() {

return packageId;

}

public void setPackageId(String packageId) {

this.packageId = packageId;

}

public LocalDate getStartDate() {

return startDate;

}

public void setStartDate(LocalDate startDate) {

this.startDate = startDate;

}

public LocalDate getEndDate() {

return endDate;

}

public void setEndDate(LocalDate endDate) {

this.endDate = endDate;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public Long getPaymentId() {

return paymentId;

}

public void setPaymentId(Long paymentId) {

this.paymentId = paymentId;

}

}**package** com.booking.entity;

**import** jakarta.persistence.\*;

@Entity

**public** **class** Payment {

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** Long paymentId;

**private** Long userId;

**private** String bookingId;

**private** **double** amount;

**private** String status;

@Column(nullable = **false**)

**private** String paymentMethod; // Should be "Credit Card" or "Debit Card"

**private** String currency;

@Transient

**private** String cardNumber; // Not stored in DB

@Transient

**private** String cvv; // Not stored in DB

@Transient

**private** String atmPin; // Not stored in DB

@Transient

**private** String expiryDate; // Not stored in DB

// Getters and Setters

**public** Long getUserId() {

**return** userId;

}

**public** **void** setUserId(Long userId) {

**this**.userId = userId;

}

**public** String getBookingId() {

**return** bookingId;

}

**public** **void** setBookingId(String bookingId) {

**this**.bookingId = bookingId;

}

**public** **double** getAmount() {

**return** amount;

}

**public** **void** setAmount(**double** amount) {

**this**.amount = amount;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

**public** String getPaymentMethod() {

**return** paymentMethod;

}

**public** **void** setPaymentMethod(String paymentMethod) {

**this**.paymentMethod = paymentMethod;

}

**public** String getCurrency() {

**return** currency;

}

**public** **void** setCurrency(String currency) {

**this**.currency = currency;

}

**public** String getCardNumber() {

**return** cardNumber;

}

**public** **void** setCardNumber(String cardNumber) {

**this**.cardNumber = cardNumber;

}

**public** String getCvv() {

**return** cvv;

}

**public** **void** setCvv(String cvv) {

**this**.cvv = cvv;

}

**public** String getAtmPin() {

**return** atmPin;

}

**public** **void** setAtmPin(String atmPin) {

**this**.atmPin = atmPin;

}

**public** Long getPaymentId() {

**return** paymentId;

}

**public** **void** setPaymentId(Long paymentId) {

**this**.paymentId = paymentId;

}

**public** String getExpiryDate() {

**return** expiryDate;

}

**public** **void** setExpiryDate(String expiryDate) {

**this**.expiryDate = expiryDate;

}

}

package com.booking.repository;

import com.booking.entity.Booking;

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

import java.util.List;

public interface BookingRepository extends JpaRepository<Booking, Long> {

List<Booking> findByUserId(Long userId);

int countByUserId(Long userId);

List<Booking> findByPackageId(String packageId);

}package com.booking.repository;

import com.booking.entity.Payment;

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

public interface PaymentRepository extends JpaRepository<Payment, Long> {

}package com.booking.response;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class ApiResponse {

private boolean success;

private String message;

private Object data;

}**package** com.booking.service;

**import** com.fasterxml.jackson.databind.ObjectMapper;

**import** com.booking.client.TravelPackageClient;

**import** com.booking.DTO.TravelPackageDTO;

**import** com.booking.entity.Booking;

**import** com.booking.repository.BookingRepository;

**import** com.booking.response.ApiResponse;

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

**import** org.springframework.http.ResponseEntity;

**import** org.springframework.stereotype.Service;

**import** java.time.LocalDate;

**import** java.util.List;

**import** java.util.stream.Collectors;

@Service

**public** **class** BookingService {

@Autowired

**private** ObjectMapper objectMapper;

@Autowired

**private** BookingRepository bookingRepo;

**private** **final** TravelPackageClient travelPackageClient;

@Autowired

**public** BookingService(TravelPackageClient travelPackageClient) {

**this**.travelPackageClient = travelPackageClient;

}

**public** Booking createBooking(Booking booking) {

ApiResponse response = travelPackageClient.getPackageById(Long.*parseLong*(booking.getPackageId()));

Object data = response.getData();

// Convert the response data to TravelPackageDTO

TravelPackageDTO travelPackage = objectMapper.convertValue(data, TravelPackageDTO.**class**);

**if** (travelPackage == **null**) {

**throw** **new** IllegalArgumentException("Invalid package ID");

}

booking.setStatus("CONFIRMED");

**return** bookingRepo.save(booking);

}

**public** List<Booking> getAllBookings() {

**return** bookingRepo.findAll();

}

**public** Booking getBookingById(Long id) {

**return** bookingRepo.findById(id).orElse(**null**);

}

**public** **void** deleteBooking(Long id) {

bookingRepo.deleteById(id);

}

**public** ResponseEntity<String> cancelBooking(Long bookingId) {

Booking booking = bookingRepo.findById(bookingId).orElse(**null**);

**if** (booking == **null**) {

**return** ResponseEntity.*badRequest*().body("Booking not found.");

}

LocalDate today = LocalDate.*now*();

**if** (booking.getStartDate().minusDays(7).isBefore(today)) {

**return** ResponseEntity.*badRequest*().body("Cancellation not allowed. Must cancel at least 7 days before departure.");

}

booking.setStatus("CANCELLED");

bookingRepo.save(booking);

**return** ResponseEntity.*ok*("Booking cancelled successfully.");

}

**private** List<TravelPackageDTO> castToTravelPackageDTOList(Object data) {

**if** (data **instanceof** List<?>) {

**return** ((List<?>) data).stream()

.filter(item -> item **instanceof** TravelPackageDTO)

.map(item -> (TravelPackageDTO) item)

.collect(Collectors.*toList*());

}

**return** List.*of*();

}

**public** **int** getBookingCountByUser(Long userId) {

**return** bookingRepo.countByUserId(userId);

}

**public** List<Booking> getBookingsByUser(Long userId) {

**return** bookingRepo.findByUserId(userId);

}

**public** List<Booking> getBookingsByPackageId(String packageId) {

**return** bookingRepo.findByPackageId(packageId);

}

}

package com.booking.service;

import com.booking.entity.Booking;

import com.booking.entity.Payment;

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

import org.springframework.mail.SimpleMailMessage;

import org.springframework.mail.javamail.JavaMailSender;

import org.springframework.stereotype.Service;

@Service

public class NotificationService {

@Autowired

private JavaMailSender mailSender;

private final String senderEmail = "indhaanman@gmail.com";

private final String customerEmail = "snehithkumar535@gmail.com";

private final String agentEmail = "padmajareddy630@gmail.com";

//Booking details are sent to customers through Email

public void notifyCustomer(Booking booking, Payment payment) {

String subject = "Your Travel Booking is Confirmed – Booking ID: " + booking.getBookingId();

String body = "Dear Customer,\n\n"

+ "Thank you for booking your travel with us!\n\n"

+ "Your booking has been successfully confirmed. Here are the details:\n\n"

+ "- Booking ID: " + booking.getBookingId() + "\n"

+ "- Package ID: " + booking.getPackageId() + "\n"

+ "- Travel Dates: " + booking.getStartDate() + " to " + booking.getEndDate() + "\n"

+ "- Payment Amount: " + payment.getAmount() + " "+ payment.getCurrency() + "\n"

+ "- Payment Status: " + payment.getStatus() + "\n\n"

+ "We look forward to providing you with a wonderful travel experience.\n\n"

+ "Warm regards,\nTravel Booking Team";

sendEmail(customerEmail, subject, body);

System.out.println("Email is sent to " + customerEmail);

}

//Booking details are sent to travel agent through Email

public void notifyTravelAgent(Booking booking, Payment payment) {

String subject = "New Booking Received – Booking ID: " + booking.getBookingId();

String body = "Dear Travel Agent,\n\n"

+ "A new booking has been successfully made. Please find the details below:\n\n"

+ "- Booking ID: " + booking.getBookingId() + "\n"

+ "- Customer ID: " + booking.getUserId() + "\n"

+ "- Package ID: " + booking.getPackageId() + "\n"

+ "- Travel Dates: " + booking.getStartDate() + " to " + booking.getEndDate() + "\n"

+ "- Payment Amount: " + payment.getAmount() + " " + payment.getCurrency() + "\n"

+ "- Payment Status: " + payment.getStatus() + "\n\n"

+ "Please ensure all arrangements are in place for the customer's travel.\n\n"

+ "Best regards,\nTravel Booking System";

sendEmail(agentEmail, subject, body);

System.out.println("Email is sent to " + agentEmail);

}

private void sendEmail(String to, String subject, String body) {

SimpleMailMessage mail = new SimpleMailMessage();

mail.setFrom(senderEmail);

mail.setTo(to);

mail.setSubject(subject);

mail.setText(body);

mailSender.send(mail);

}

}package com.booking.service;

import com.booking.entity.Booking;

import com.booking.entity.Payment;

import com.booking.repository.BookingRepository;

import com.booking.repository.PaymentRepository;

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

import org.springframework.stereotype.Service;

import java.time.LocalDate;

import java.util.List;

@Service

public class PaymentService {

@Autowired

private PaymentRepository paymentRepo;

@Autowired

private BookingRepository bookingRepo;

@Autowired

private NotificationService notificationService;

public Payment processPayment(Payment payment) {

if (!payment.getPaymentMethod().equalsIgnoreCase("Credit Card") &&

!payment.getPaymentMethod().equalsIgnoreCase("Debit Card")) {

throw new IllegalArgumentException("Only Credit Card or Debit Card are accepted.");

}

if (payment.getCardNumber() == null || !payment.getCardNumber().matches("\\d{16}")) {

throw new IllegalArgumentException("Invalid card number. Must be 16 digits.");

}

if (payment.getCvv() == null || !payment.getCvv().matches("\\d{3}")) {

throw new IllegalArgumentException("Invalid CVV. Must be 3 digits.");

}

if (payment.getAtmPin() == null || !payment.getAtmPin().matches("\\d{4}")) {

throw new IllegalArgumentException("Invalid ATM PIN. Must be 4 digits.");

}

// Expiry Date Format: MM/YY

if (payment.getExpiryDate() == null || !payment.getExpiryDate().matches("\\d{2}/\\d{2}")) {

throw new IllegalArgumentException("Expiry date must be in MM/YY format.");

}

String[] parts = payment.getExpiryDate().split("/");

int expMonth = Integer.parseInt(parts[0]);

int expYear = 2000 + Integer.parseInt(parts[1]); // e.g., "25" becomes 2025

if (expMonth < 1 || expMonth > 12) {

throw new IllegalArgumentException("Expiry month must be between 01 and 12.");

}

int currentMonth = LocalDate.now().getMonthValue();

int currentYear = LocalDate.now().getYear();

if (expYear < currentYear || (expYear == currentYear && expMonth < currentMonth)) {

throw new IllegalArgumentException("Card has expired.");

}

payment.setStatus("PAID");

Payment savedPayment = paymentRepo.save(payment);

// Fetch booking details using bookingId

Long bookingId = Long.parseLong(payment.getBookingId());

Booking booking = bookingRepo.findById(bookingId).orElse(null);

if (booking != null) {

notificationService.notifyCustomer(booking, savedPayment);

notificationService.notifyTravelAgent(booking, savedPayment);

}

return savedPayment;

}

public List<Payment> getAllPayments() {

return paymentRepo.findAll();

}

public Payment getPaymentById(Long id) {

return paymentRepo.findById(id).orElse(null);

}

}spring.application.name=TravelBooking\_PaymentModule

spring.datasource.url=jdbc:mysql://localhost:3306/travel\_db

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

logging.level.org.hibernate.SQL=debug

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

server.port=8086

spring.mail.host=smtp.gmail.com

spring.mail.port=587

spring.mail.username=indhaanman@gmail.com

spring.mail.password=ibzbwveckpfjibey

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

spring.cloud.compatibility-verifier.enabled=false

# Eureka Client Configuration

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

eureka.instance.prefer-ip-address=true

this is all files related to the booking and payment module   
I want to implement microservices for this module from package module using feign client search packages ,booking logic present and payment in this module notification also implement microservices correctly …with these 2 modules from package module only search through feign client