package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

System.err.println("Eureka server running on port number 8761");

}

}

Application.properties

spring.application.name=eureka-server

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

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

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication(scanBasePackages = "com")

@EnableDiscoveryClient

@EntityScan(basePackages = "com.entity")

@EnableJpaRepositories(basePackages = "com.repository")

public class CabBookingMicroServiceApplication {

public static void main(String[] args) {

SpringApplication.run(CabBookingMicroServiceApplication.class, args);

System.err.println("cab micro service up on port no 9090");

}

}

package com.entity;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import lombok.Data;

@Entity

@Data

public class Cab {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int cid;

private String typeofcab;

private int seatcapacity;

private String company;

private double fare;

}

package com.repository;

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

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

import org.springframework.stereotype.Repository;

import com.entity.Cab;

@Repository

public interface CabRepository extends JpaRepository<Cab, Integer>{

@Query("SELECT c.fare FROM Cab c WHERE c.cid = ?1")

Double findFareByCabId(int cid);

}

package com.service;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.entity.Cab;

import com.repository.CabRepository;

@Service

public class CabService {

@Autowired

CabRepository cabRepository;

public String storeCab(Cab cab) {

cabRepository.save(cab);

return "Account Created";

}

public String updateCab(Cab cab) {

Optional<Cab> result=cabRepository.findById(cab.getCid());

if(result.isPresent()) {

Cab c=result.get();

c.setTypeofcab(cab.getTypeofcab());

c.setSeatcapacity(cab.getSeatcapacity());

c.setFare(cab.getFare());

cabRepository.saveAndFlush(c);

return "Cab Information Updated Successfully";

}else {

return "Cab Record Not Present";

}

}

public String deletecab(int cid) {

Optional<Cab> result=cabRepository.findById(cid);

if(result.isPresent()) {

cabRepository.deleteById(cid);

return "Cab Record Deleted Successfully";

}else {

return "Cab Record Not Present";

}

}

public List<Cab> findAllCabs(){

return cabRepository.findAll();

}

public Cab findCabById(int cid) {

Optional<Cab> result=cabRepository.findById(cid);

if(result.isPresent()) {

Cab c=result.get();

return c;

}else {

return null;

}

}

public Double findFareByCabId(int cid) {

return cabRepository.findFareByCabId(cid);

}

}

package com.controller;

import java.util.List;

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

import org.springframework.http.MediaType;

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 com.entity.Cab;

import com.service.CabService;

@RestController

@RequestMapping("travel") // http://localhost:9090/travel/\*

public class CabController {

@Autowired

CabService cabService;

@PostMapping(value = "store\_cab",consumes = MediaType.APPLICATION\_JSON\_VALUE)

public String storeCab(@RequestBody Cab cab) {

return cabService.storeCab(cab);

}

@PutMapping(value="update\_cab",consumes = MediaType.APPLICATION\_JSON\_VALUE)

public String updateCab(@RequestBody Cab cab) {

return cabService.updateCab(cab);

}

@DeleteMapping(value="delete\_cab/{cid}")

public String deleteCab(@PathVariable("cid") int cid) {

return cabService.deletecab(cid);

}

@GetMapping(value="find\_cabs",consumes = MediaType.APPLICATION\_JSON\_VALUE)

public List<Cab> findAllCabs(){

return cabService.findAllCabs();

}

@GetMapping(value="find\_cab\_by\_id/{id}",consumes = MediaType.APPLICATION\_JSON\_VALUE)

public Cab findCabById(@PathVariable("id") int cid) {

return cabService.findCabById(cid);

}

@GetMapping(value = "find\_fare\_by\_cab\_id/{cid}")

public Double findFareByCabId(@PathVariable("cid") int cid) {

return cabService.findFareByCabId(cid);

}

}

Application.properties

spring.application.name=cab-booking-micro-service

server.port=9090

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

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

spring.datasource.username=root

spring.datasource.password=saivarun

spring.jpa.hibernate.ddl-auto=update

package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.autoconfigure.domain.EntityScan;

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

import org.springframework.context.annotation.Bean;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import org.springframework.web.client.RestTemplate;

@SpringBootApplication(scanBasePackages = "com")

@EnableDiscoveryClient

@EntityScan(basePackages = "com.entity")

@EnableJpaRepositories(basePackages = "com.repository")

public class BookingMicroServiceApplication {

public static void main(String[] args) {

SpringApplication.run(BookingMicroServiceApplication.class, args);

System.err.println("Booking micro service up on port no 9191");

}

@Bean

public RestTemplate restTemplate() {

return new RestTemplate();

}

}

package com.entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import lombok.Data;

@Entity

@Data

public class Booking {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int bid;

private String origin;

private String destination;

private double distance;

private double amount;

@Column(nullable = true)

private int cid;

}

package com.repository;

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

import org.springframework.stereotype.Repository;

import com.entity.Booking;

@Repository

public interface BookingRepository extends JpaRepository<Booking, Integer>{

}

package com.service;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import org.springframework.web.client.RestTemplate;

import com.entity.Booking;

import com.repository.BookingRepository;

@Service

public class BookingService {

@Autowired

BookingRepository bookingRepository;

@Autowired

RestTemplate restTemplate;

// Method to add booking and calculate fare based on cab info

public String addBooking(Booking booking) {

// Call Cab Microservice to get the cab fare using cid

String fareUrl = "http://localhost:9090/travel/find\_fare\_by\_cab\_id/" + booking.getCid();

Double fare = restTemplate.getForObject(fareUrl, Double.class);

if (fare != null) {

// Calculate the fare based on distance and cab fare per kilometer

double totalFare = booking.getDistance() \* fare;

booking.setAmount(totalFare);

bookingRepository.save(booking);

return "Booking details saved. Total fare: " + totalFare;

} else {

return "Cab with ID " + booking.getCid() + " not found.";

}

}

// Method to find a booking by its ID

public Optional<Booking> findBookingById(int bid) {

return bookingRepository.findById(bid);

}

// Method to retrieve cab details by booking ID

public String findCabId(int bid) {

Optional<Booking> result = bookingRepository.findById(bid);

if (result.isPresent()) {

Booking booking = result.get();

String data = restTemplate.getForObject("http://localhost:9090/travel/find\_cab\_by\_id/" + booking.getCid(), String.class);

return data;

} else {

return "Data not present";

}

}

}

package com.controller;

import java.util.Optional;

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

import org.springframework.http.MediaType;

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

import com.entity.Booking;

import com.service.BookingService;

@RestController

@RequestMapping("booking") // Base URL for all booking-related endpoints

public class BookingController {

@Autowired

BookingService bookingService;

// Endpoint to add a new booking

@PostMapping(value = "/add\_booking", consumes = MediaType.APPLICATION\_JSON\_VALUE)

public String addBooking(@RequestBody Booking booking) {

return bookingService.addBooking(booking);

}

// Endpoint to find a booking by its ID

@GetMapping(value = "/find\_booking\_by\_id/{bid}", produces = MediaType.APPLICATION\_JSON\_VALUE)

public Optional<Booking> findBookingById(@PathVariable("bid") int bid) {

return bookingService.findBookingById(bid);

}

// Endpoint to retrieve cab details by booking ID

@GetMapping(value = "/find\_cab\_by\_booking\_id/{bid}", produces = MediaType.APPLICATION\_JSON\_VALUE)

public String findCabId(@PathVariable("bid") int bid) {

return bookingService.findCabId(bid);

}

}

spring.application.name=booking-micro-service

server.port=9191

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

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

spring.datasource.username=root

spring.datasource.password=saivarun

spring.jpa.hibernate.ddl-auto=update