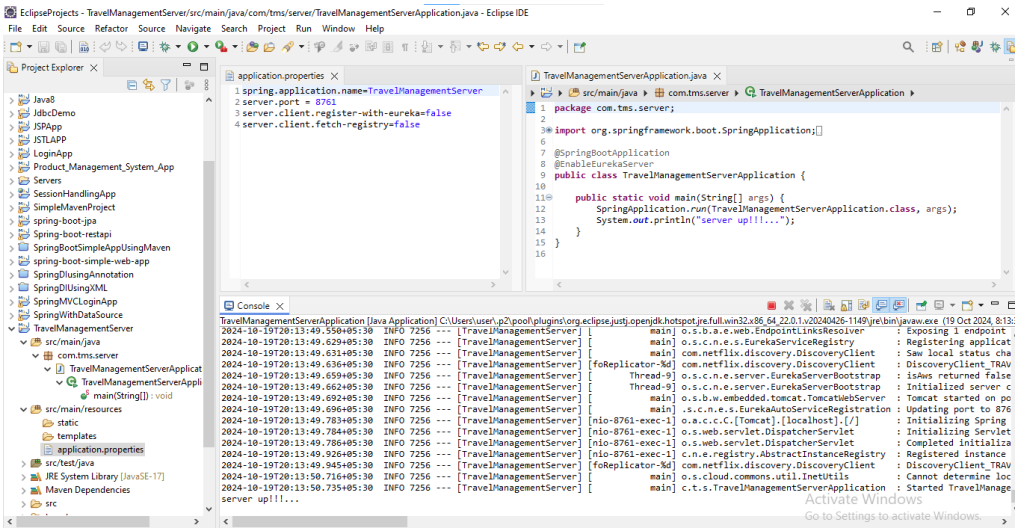
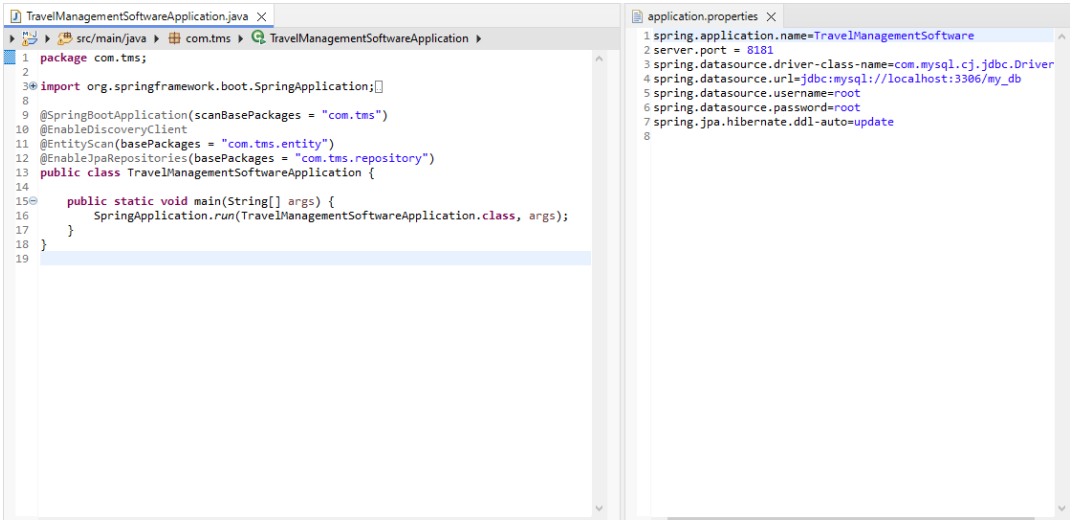
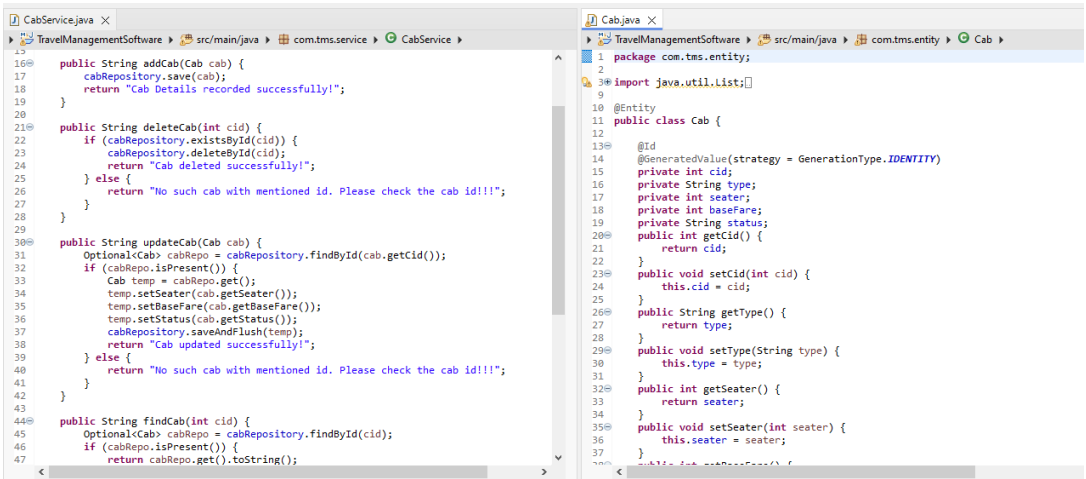
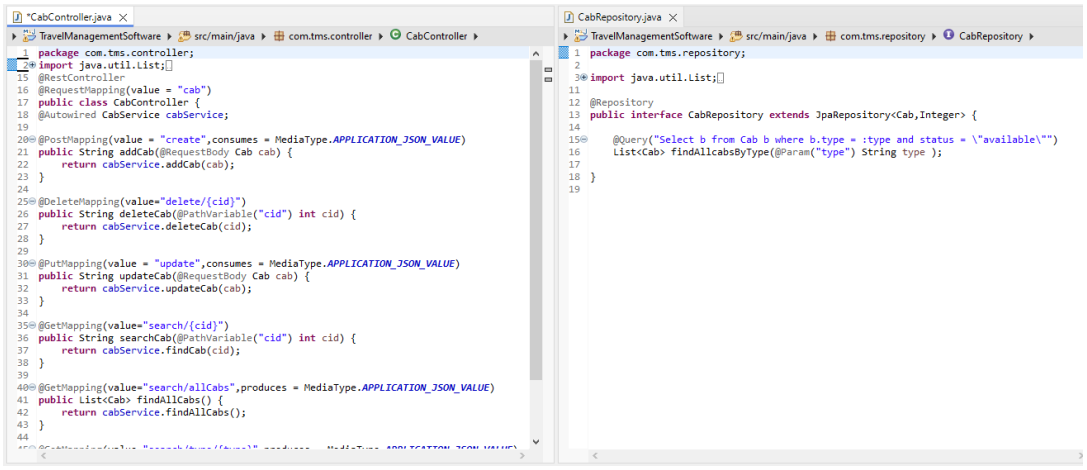

EclipseProjects - TravelManagementServer/src/main/java/com/tms/server/TravelManagementServerApplication.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project Explorer
application.properties
spring.application.name=TravelManagementServer
server.port = 8761
server.client.register-with-eureka=false
server.client.fetch-registry=false
TravelManagementServerApplication.java
package com.tms.server;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
@EnableEurekaServer
public class TravelManagementServerApplication {
public static void main(String[] args) {
SpringApplication.run(TravelManagementServerApplication.class, args);
System.out.println("server up!!!...");
}
}
Console
server up!!!...
Activate Windows
Go to Settings to activate Windows.


TravelManagementSoftwareApplication.java
src/main/java com.tms TravelManagementSoftwareApplication
package com.tms;
import org.springframework.boot.SpringApplication;
@SpringBootApplication(scanBasePackages = "com.tms")
@EnableDiscoveryClient
@EntityScan(basePackages = "com.tms.entity")
@EnableJpaRepositories(basePackages = "com.tms.repository")
public class TravelManagementSoftwareApplication {
public static void main(String[] args) {
SpringApplication.run(TravelManagementSoftwareApplication.class, args);
}
}
application.properties
spring.application.name=TravelManagementSoftware
server.port = 8181
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/my_db
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update


CabService.java
TravelManagementSoftware src/main/java com.tms.service CabService
public String addCab(Cab cab) {
cabRepository.save(cab);
return "Cab Details recorded successfully!";
}
public String deleteCab(int cid) {
if (cabRepository.existsById(cid)) {
cabRepository.deleteById(cid);
return "Cab deleted successfully!";
} else {
return "No such cab with mentioned id. Please check the cab id!!!";
}
}
public String updateCab(Cab cab) {
Optional<Cab> cabRepo = cabRepository.findById(cab.getCid());
if (cabRepo.isPresent()) {
Cab temp = cabRepo.get();
temp.setSeater(cab.getSeater());
temp.setBaseFare(cab.getBaseFare());
temp.setStatus(cab.getStatus());
cabRepository.saveAndFlush(temp);
return "Cab updated successfully!";
} else {
return "No such cab with mentioned id. Please check the cab id!!!";
}
}
public String findCab(int cid) {
Optional<Cab> cabRepo = cabRepository.findById(cid);
if (cabRepo.isPresent()) {
return cabRepo.get().toString();
Cab.java
TravelManagementSoftware src/main/java com.tms.entity Cab
package com.tms.entity;
import java.util.List;
@Entity
public class Cab {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int cid;
private String type;
private int seater;
private int baseFare;
private String status;
public int getCid() {
return cid;
}
public void setCid(int cid) {
this.cid = cid;
}
public String getType() {
return type;
}
public void setType(String type) {
this.type = type;
}
public int getSeater() {
return seater;
}
public void setSeater(int seater) {
this.seater = seater;
}



*CabController.java

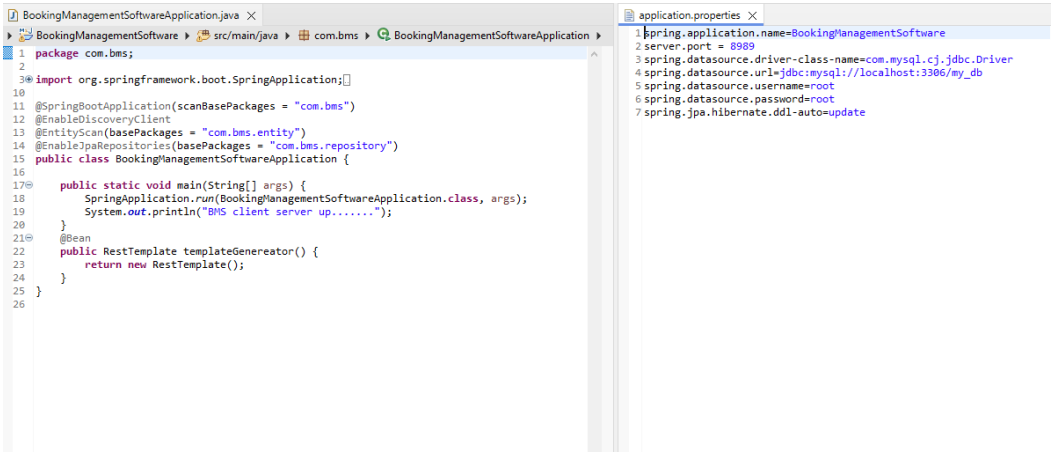
TravelManagementSoftware ▸ src/main/java ▸ com.tms.controller ▸ CabController ▸

```java
package com.tms.controller;
import java.util.List;
@RestController
@RequestMapping(value = "cab")
public class CabController {
@Autowired CabService cabService;

@PostMapping(value = "create",consumes = MediaType.APPLICATION_JSON_VALUE)
public String addCab(@RequestBody Cab cab) {
    return cabService.addCab(cab);
}

@DeleteMapping(value="delete/{cid}")
public String deleteCab(@PathVariable("cid") int cid) {
    return cabService.deleteCab(cid);
}

@PutMapping(value = "update",consumes = MediaType.APPLICATION_JSON_VALUE)
public String updateCab(@RequestBody Cab cab) {
    return cabService.updateCab(cab);
}

@GetMapping(value="search/{cid}")
public String searchCab(@PathVariable("cid") int cid) {
    return cabService.findCab(cid);
}

@GetMapping(value="search/allCabs",produces = MediaType.APPLICATION_JSON_VALUE)
public List<Cab> findAllCabs() {
    return cabService.findAllCabs();
}
```

CabRepository.java

TravelManagementSoftware ▸ src/main/java ▸ com.tms.repository ▸ CabRepository ▸

```java
package com.tms.repository;

import java.util.List;

@Repository
public interface CabRepository extends JpaRepository<Cab,Integer> {

    @Query("Select b from Cab b where b.type = :type and status = \"available\"")
    List<Cab> findAllcabsByType(@Param("type") String type );

}
```



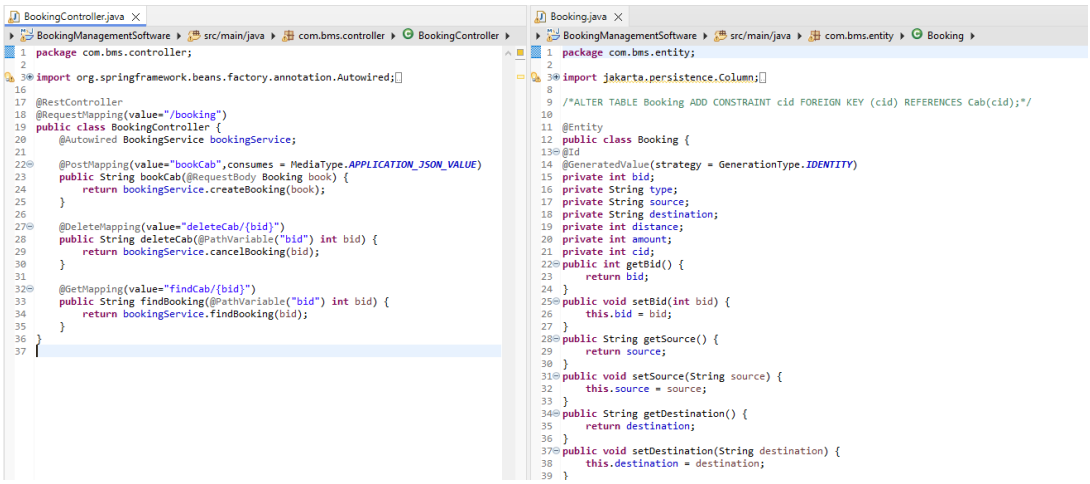
BookingManagementSoftwareApplication.java

BookingManagementSoftware ▸ src/main/java ▸ com.bms ▸ BookingManagementSoftwareApplication ▸

```java
package com.bms;

import org.springframework.boot.SpringApplication;

@SpringBootApplication(scanBasePackages = "com.bms")
@EnableDiscoveryClient
@EntityScan(basePackages = "com.bms.entity")
@EnableJpaRepositories(basePackages = "com.bms.repository")
public class BookingManagementSoftwareApplication {

    public static void main(String[] args) {
        SpringApplication.run(BookingManagementSoftwareApplication.class, args);
        System.out.println("BMS client server up.......");
    }
    @Bean
    public RestTemplate templateGenereator() {
        return new RestTemplate();
    }
}
```

application.properties

```
spring.application.name=BookingManagementSoftware
server.port = 8989
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/my_db
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
```



BookingController.java

BookingManagementSoftware ▸ src/main/java ▸ com.bms.controller ▸ BookingController ▸

```java
package com.bms.controller;

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

@RestController
@RequestMapping(value="/booking")
public class BookingController {
    @Autowired BookingService bookingService;

    @PostMapping(value="bookCab",consumes = MediaType.APPLICATION_JSON_VALUE)
    public String bookCab(@RequestBody Booking book) {
        return bookingService.createBooking(book);
    }

    @DeleteMapping(value="deleteCab/{bid}")
    public String deleteCab(@PathVariable("bid") int bid) {
        return bookingService.cancelBooking(bid);
    }

    @GetMapping(value="findCab/{bid}")
    public String findBooking(@PathVariable("bid") int bid) {
        return bookingService.findBooking(bid);
    }
}
```

Booking.java

BookingManagementSoftware ▸ src/main/java ▸ com.bms.entity ▸ Booking ▸

```java
package com.bms.entity;

import jakarta.persistence.Column;

/*ALTER TABLE Booking ADD CONSTRAINT cid FOREIGN KEY (cid) REFERENCES Cab(cid);*/

@Entity
public class Booking {
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int bid;
private String type;
private String source;
private String destination;
private int distance;
private int amount;
private int cid;
public int getBid() {
    return bid;
}
public void setBid(int bid) {
    this.bid = bid;
}
public String getSource() {
    return source;
}
public void setSource(String source) {
    this.source = source;
}
public String getDestination() {
    return destination;
}
public void setDestination(String destination) {
    this.destination = destination;
}
```

BookingService.java

BookingManagementSoftware ▸ src/main/java ▸ com.bms.service ▸ BookingService

```java
package com.bms.service;

import java.util.List;

@Service
public class BookingService {
@Autowired BookingRepository bookingRepository;
@Autowired RestTemplate restTemplate;

public String createBooking(Booking book) {
    Map<String, Object> response = restTemplate.getForObject("http://localhos
    int Totalamount = book.getDistance()*(Integer)(response.get("baseFare"));
    book.setAmount(Totalamount);
    book.setCid((Integer)(response.get("cid")));
    bookingRepository.save(book);
    return "Booking successfull!";
}
public String cancelBooking(int bid) {
    if (bookingRepository.existsById(bid)) {
        bookingRepository.deleteById(bid);
        return "Booking cancelled";
    } else {
        return "Can't find the booking. Check the booking id!!!";
    }
}
public String findBooking(int bid) {
    if (bookingRepository.existsById(bid)) {
        return bookingRepository.findById(bid).get().toString();
    } else {
        return "Can't find the booking. Check the booking id!!!";
    }
}
}

```

BookingRepository.java

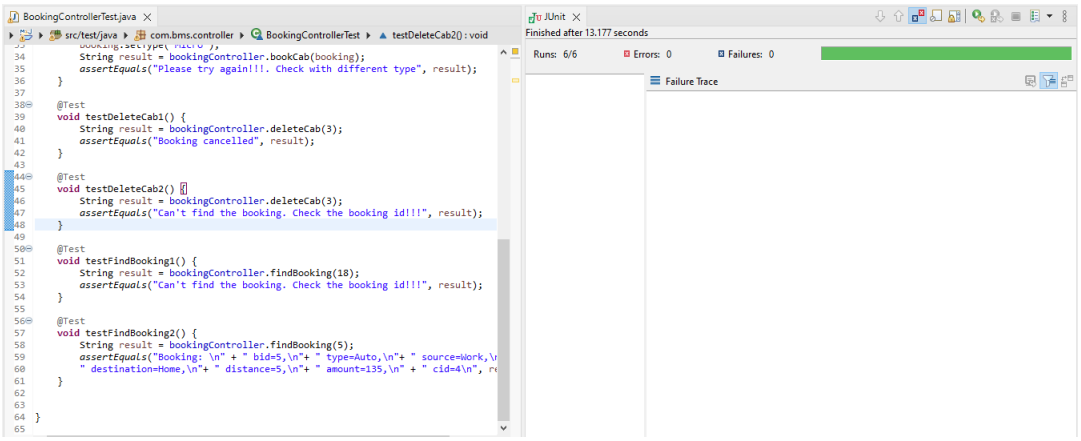
BookingManagementSoftware ▸ src/main/java ▸ com.bms.repository ▸ BookingRepository

```java
package com.bms.repository;

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

public interface BookingRepository extends JpaRepository<Booking,Integer> {

}

```

BookingControllerTest.java

BookingManagementSoftware ▸ src/test/java ▸ com.bms.controller ▸ BookingControllerTest ▸ testDeleteCab2() : void

```java
package com.bms.controller;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;

import com.bms.entity.Booking;

@SpringBootTest
class BookingControllerTest {
    @Autowired BookingController bookingController;

    @Test
    void testBookCab1() {
        Booking booking = new Booking();
        booking.setDestination("Home");
        booking.setDistance(5);
        booking.setSource("Work");
        booking.setType("Auto");
        String result = bookingController.bookCab(booking);
        assertEquals("Booking successfull!", result);
    }

    @Test
    void testBookCab2() {
        Booking booking = new Booking();
        booking.setDestination("Home");
        booking.setDistance(5);
        booking.setSource("Work");
        booking.setType("Micro");
        String result = bookingController.bookCab(booking);
        assertEquals("Please try again!!!. Check with different type", result);
    }

    @Test
    void testDeleteCab1() {
        String result = bookingController.deleteCab(3);
        assertEquals("Booking cancelled", result);
    }

    @Test
    void testDeleteCab2() {
        String result = bookingController.deleteCab(3);
        assertEquals("Can't find the booking. Check the booking id!!!", result);
    }

    @Test
    void testFindBooking1() {
        String result = bookingController.findBooking(18);
        assertEquals("Can't find the booking. Check the booking id!!!", result);
    }

    @Test
    void testFindBooking2() {
        String result = bookingController.findBooking(5);
        assertEquals("Booking: \n" + " bid=5,\n"+ " type=Auto,\n"+ " source=Work,\n"+
        " destination=Home,\n"+ " distance=5,\n"+ " amount=135,\n" + " cid=4\n", result);
    }
}

```


BookingControllerTest.java
src/test/java
com.bms.controller
BookingControllerTest
testDeleteCab2() : void
booking.setType("AUTO");
String result = bookingController.bookCab(booking);
assertEquals("Please try again!!!. Check with different type", result);
}
@Test
void testDeleteCab1() {
String result = bookingController.deleteCab(3);
assertEquals("Booking cancelled", result);
}
@Test
void testDeleteCab2() {
String result = bookingController.deleteCab(3);
assertEquals("Can't find the booking. Check the booking id!!!", result);
}
@Test
void testFindBooking1() {
String result = bookingController.findBooking(18);
assertEquals("Can't find the booking. Check the booking id!!!", result);
}
@Test
void testFindBooking2() {
String result = bookingController.findBooking(5);
assertEquals("Booking: \n" + " bid=5,\n"+ " type=Auto,\n"+ " source=Work,\n
" destination=Home,\n"+ " distance=5,\n"+ " amount=135,\n" + " cid=4\n", re
}
}
JUnit
Finished after 13.177 seconds
Runs: 6/6
Errors: 0
Failures: 0
Failure Trace