SPRING ASSIGNMENT 2

Submitted By -

Name -Ritesh Patil

Roll No – 72288097K

Employee.java

```
package com.infosys.orm.entity;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;


@Data

@AllArgsConstructor

@NoArgsConstructor

@Entity

public class Employee {


	@Id

	@Column(name = "emp_id")

	private Integer empId;

	@Column(name = "emp_name")

	private String empName;

	@Column(name = "emp_dept")

	private String dept;

	@Column(name = "base_location")

	private String baseLocation;
```

```
    private Integer address;


    @Column(name = "emp_salary")
    private Double empSalary;
    @Column(name = "emp_bandlevel")
    private String empBandlevel;
    @Column(name = "contact_no")
    private String contactNo;



    public static EmployeeDTO createEmployeeDTO(Employee emp) {
        return new EmployeeDTO(emp.getEmpId(), emp.getEmpName(), emp.getDept(), emp.getBaseLocation(), emp.getAddress(), emp.getEmpSalary(), emp.getEmpBandlevel(), emp.getContactNo());
    }
}
```

EmployeeRepository.java

```
package com.infosys.orm.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import com.infosys.orm.entity.Employee;
public interface EmployeeRepository extends JpaRepository<Employee, Integer>{
    List<Employee> findByEmpBandlevel(String bandlevel);
    List<Employee> findByEmpName(String empName);
    List<Employee> findByEmpSalaryBetween(double s1, double s2);
    List<Employee> findByEmpBandlevelAndDept(String bandlevel, String empDept);
}
```

EmployeeService.java

```
package com.infosys.orm.service;
```

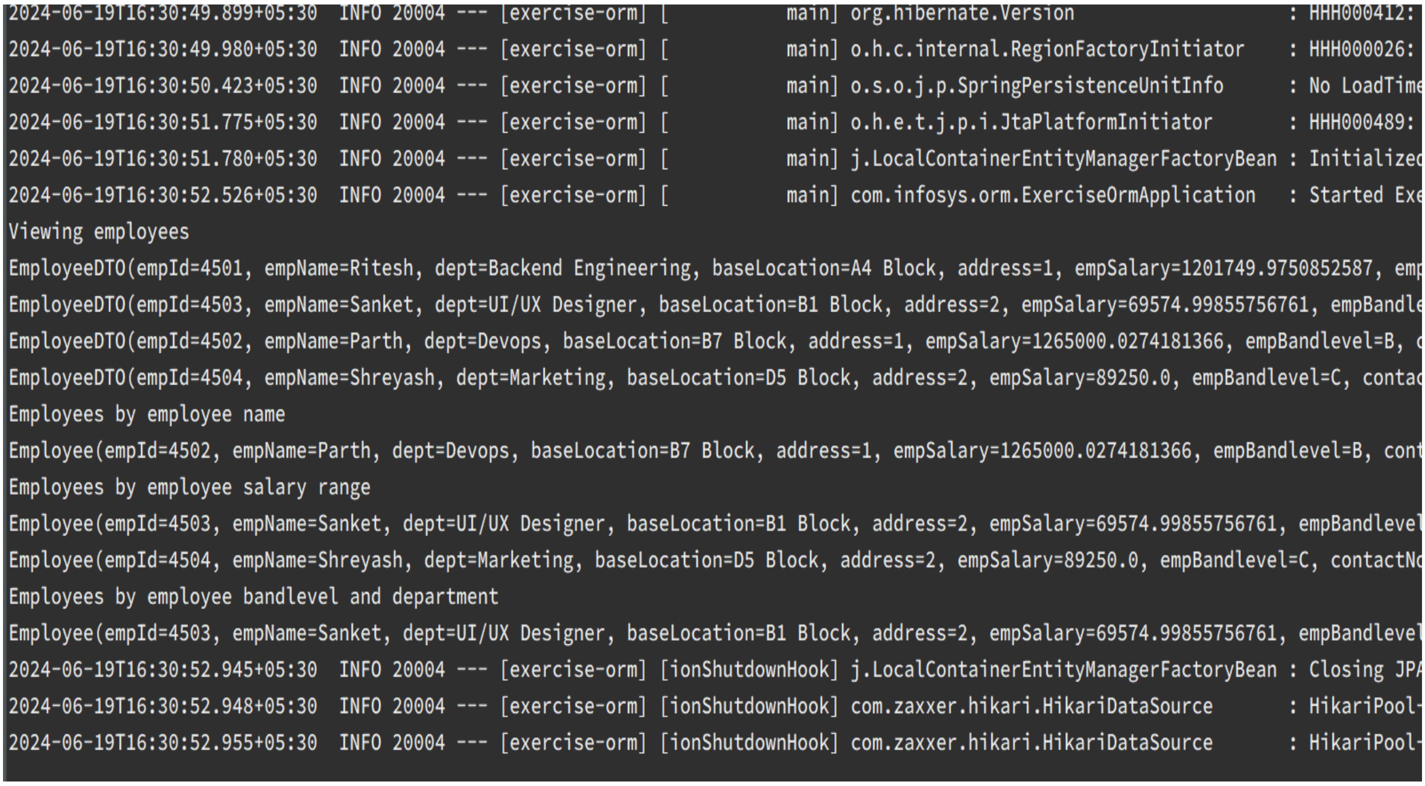
```
import java.util.List;
import com.infosys.orm.entity.Employee;
import com.infosys.orm.entity.EmployeeDTO;

public interface EmployeeService {
	public void insertEmployee(EmployeeDTO empDTO);
	public EmployeeDTO searchEmployee(Integer id);
	public List<EmployeeDTO> viewEmployees();
	public void editEmployee(Integer id, String dept);
	public void removeEmployee(Integer id);
	public List<Employee> getByBandlevel(String bandlevel);
	public List<Employee> getByEmpName(String empName);
	public List<Employee> getByEmpSalaryBetween(double s1, double s2);
	public List<Employee> getByEmpBandlevelAndEmpDept(String bandlevel, String empDept);
}
```

Main.java

```
package com.infosys.orm;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import com.infosys.orm.entity.Employee;
import com.infosys.orm.entity.EmployeeDTO;
import com.infosys.orm.repository.EmployeeRepository;
import com.infosys.orm.service.EmployeeService;

@SpringBootApplication
public class ExerciseOrmApplication implements CommandLineRunner {
	@Autowired
	EmployeeService employee;
	@Autowired
	EmployeeRepository empRepository;

	public static void main(String[] args) {
		SpringApplication.run(ExerciseOrmApplication.class, args);
	}

	@Override
	public void run(String... args) throws Exception {

//		1. INSERTING EMPLOYEE
//		System.out.println("Adding employees");
//		EmployeeDTO empDTO = new EmployeeDTO(4501, "Ritesh", "Architechture Design",
"A4 Block", 1);
//
//		try {
//			employee.insertEmployee(empDTO);
//			System.out.println("Inserted successfully");
//		}catch (Exception e) {
//			System.out.println(e.getMessage());
//		}

//		2. SEARCH EMPLOYEE
//		System.out.println("Searching employee");
//		try {
//			System.out.println(employee.searchEmployee(4501));
//		} catch (Exception e) {
//			System.out.println(e.getMessage());
//              }
//

//
//
////            4. EDITING EMPLOYEE
//              System.out.println("Editing employee details");
//              try {
//                      employee.editEmployee(4501, "Devops");
//              } catch (Exception e) {
//                      System.out.println(e.getMessage());
//              }

//              5. REMOVE EMPLOYEE
//              System.out.println("Removing employees");
//              try {
//                      employee.removeEmployee(4503);
//              } catch (Exception e) {
//                      System.out.println(e.getMessage());
//              }

//              6. GET EMPLOYEES ACCORDING TO LABEL
                System.out.println("Employees by Bandlevel");
                List<Employee> empByBandlevelA = employee.getByBandlevel("A");
//              UPDATE BY LABEL A
                for (Employee emp : empByBandlevelA) {
                        emp.setEmpSalary(emp.getEmpSalary() * 1.15f);
                        empRepository.saveAndFlush(emp);
                }
//
                System.out.println("Employees by Bandlevel");
```

```
		List<Employee> empByBandlevelB= employee.getByBandlevel("B");
//		UPDATE BY LABEL B
		for (Employee emp : empByBandlevelB) {
			emp.setEmpSalary(emp.getEmpSalary() * 1.10f);
			empRepository.saveAndFlush(emp);
		}
//
		System.out.println("Employees by Bandlevel");
		List<Employee> empByBandlevelC = employee.getByBandlevel("C");
//		UPDATE BY LABEL C
		for (Employee emp : empByBandlevelC) {
			emp.setEmpSalary(emp.getEmpSalary() * 1.05f);
			empRepository.saveAndFlush(emp);
		}

////	3.VIEW EMPLOYEES
		System.out.println("Viewing employees");
		List<EmployeeDTO> empDTOList = employee.viewEmployees();
		for (EmployeeDTO e : empDTOList) {
			System.out.println(e);
		}

//		FIND BY EMPNAME
		System.out.println("Employees by employee name");
		List<Employee> empByName = employee.getByEmpName("Parth");
		for(Employee emp : empByName) {
			System.out.println(emp);
		}

//		FIND BY EMP SALARY RANGE
		System.out.println("Employees by employee salary range");
		List<Employee> empBySalaryRange = employee.getByEmpSalaryBetween(50000, 
100000);
		for(Employee emp : empBySalaryRange) {
			System.out.println(emp);
		}


//		FIND EMP BY DEPT AND BANDLEVEL
		System.out.println("Employees by employee bandlevel and department");
		List<Employee> empByBandlevelAndDept = 
employee.getByEmpBandlevelAndEmpDept("A", "UI/UX Designer");
		for(Employee emp : empByBandlevelAndDept) {
			System.out.println(emp);
		}
	}
}
```

********************************** OUTPUT **************************************



```
2024-06-19T16:30:49.899+05:30  INFO 20004 --- [exercise-orm] [           main] org.hibernate.Version                    : HHH000412:
2024-06-19T16:30:49.980+05:30  INFO 20004 --- [exercise-orm] [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026:
2024-06-19T16:30:50.423+05:30  INFO 20004 --- [exercise-orm] [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTime
2024-06-19T16:30:51.775+05:30  INFO 20004 --- [exercise-orm] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489:
2024-06-19T16:30:51.780+05:30  INFO 20004 --- [exercise-orm] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized
2024-06-19T16:30:52.526+05:30  INFO 20004 --- [exercise-orm] [           main] com.infosys.orm.ExerciseOrmApplication   : Started Exe
Viewing employees
EmployeeDTO(empId=4501, empName=Ritesh, dept=Backend Engineering, baseLocation=A4 Block, address=1, empSalary=1201749.9750852587, emp
EmployeeDTO(empId=4503, empName=Sanket, dept=UI/UX Designer, baseLocation=B1 Block, address=2, empSalary=69574.99855756761, empBandle
EmployeeDTO(empId=4502, empName=Parth, dept=Devops, baseLocation=B7 Block, address=1, empSalary=1265000.0274181366, empBandlevel=B, c
EmployeeDTO(empId=4504, empName=Shreyash, dept=Marketing, baseLocation=D5 Block, address=2, empSalary=89250.0, empBandlevel=C, contac
Employees by employee name
Employee(empId=4502, empName=Parth, dept=Devops, baseLocation=B7 Block, address=1, empSalary=1265000.0274181366, empBandlevel=B, cont
Employees by employee salary range
Employee(empId=4503, empName=Sanket, dept=UI/UX Designer, baseLocation=B1 Block, address=2, empSalary=69574.99855756761, empBandlevel
Employee(empId=4504, empName=Shreyash, dept=Marketing, baseLocation=D5 Block, address=2, empSalary=89250.0, empBandlevel=C, contactNo
Employees by employee bandlevel and department
Employee(empId=4503, empName=Sanket, dept=UI/UX Designer, baseLocation=B1 Block, address=2, empSalary=69574.99855756761, empBandlevel
2024-06-19T16:30:52.945+05:30  INFO 20004 --- [exercise-orm] [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA
2024-06-19T16:30:52.948+05:30  INFO 20004 --- [exercise-orm] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-
2024-06-19T16:30:52.955+05:30  INFO 20004 --- [exercise-orm] [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-
```