**Create REST service to gets all employees**

1. Employee.java

package com.example.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
  
@Entity  
public class Employee {  
 @Id  
 private int id;  
 private String name;  
 private String department;  
 private double salary;  
  
 public Employee() {}  
  
 public Employee(int id, String name, String department, double salary) {  
 this.id = id;  
 this.name = name;  
 this.department = department;  
 this.salary = salary;  
 }  
  
 public int getId() { return id; }  
 public void setId(int id) { this.id = id; }  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 public String getDepartment() { return department; }  
 public void setDepartment(String department) { this.department = department; }  
  
 public double getSalary() { return salary; }  
 public void setSalary(double salary) { this.salary = salary; }  
}

2. EmployeeRepository.java

package com.example.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import com.example.model.Employee;  
  
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}

3. EmployeeService.java

package com.example.service;  
  
import java.util.List;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import com.example.model.Employee;  
import com.example.repository.EmployeeRepository;  
  
@Service  
public class EmployeeService {  
 private final EmployeeRepository employeeRepository;  
  
 public EmployeeService(EmployeeRepository employeeRepository) {  
 this.employeeRepository = employeeRepository;  
 }  
  
 @Transactional  
 public List<Employee> getAllEmployees() {  
 return employeeRepository.findAll();  
 }  
}

4. EmployeeController.java

package com.example.controller;  
  
import java.util.List;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.example.model.Employee;  
import com.example.service.EmployeeService;  
  
@RestController  
public class EmployeeController {  
 private final EmployeeService employeeService;  
  
 public EmployeeController(EmployeeService employeeService) {  
 this.employeeService = employeeService;  
 }  
  
 @GetMapping("/employees")  
 public List<Employee> getAllEmployees() {  
 return employeeService.getAllEmployees();  
 }  
}

5. EmployeeApplication.java

package com.example;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class EmployeeApplication {  
 public static void main(String[] args) {  
 SpringApplication.run(EmployeeApplication.class, args);  
 }  
}

6. DataLoader.java

package com.example.config;  
  
import org.springframework.boot.CommandLineRunner;  
import org.springframework.stereotype.Component;  
import com.example.model.Employee;  
import com.example.repository.EmployeeRepository;  
  
@Component  
public class DataLoader implements CommandLineRunner {  
 private final EmployeeRepository employeeRepository;  
  
 public DataLoader(EmployeeRepository employeeRepository) {  
 this.employeeRepository = employeeRepository;  
 }  
  
 @Override  
 public void run(String... args) throws Exception {  
 employeeRepository.save(new Employee(1, "Alice", "HR", 40000));  
 employeeRepository.save(new Employee(2, "Bob", "IT", 50000));  
 employeeRepository.save(new Employee(3, "Charlie", "Finance", 60000));  
 }  
}

application.properties

# MySQL DB Configuration  
spring.datasource.url=jdbc:mysql://localhost:3306/employeedb  
spring.datasource.username=root  
spring.datasource.password=root  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
  
# JPA Properties  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

pom.xml Dependency

<dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <scope>runtime</scope>  
</dependency>

Output





