**Hands on 4: Difference between JPA, Hibernate and Spring Data JPA** 

**Java Persistence API (JPA)**

* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementation of JPA

**Hibernate**

* ORM Tool that implements JPA

**Spring Data JPA**

* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transactions

**Solution:**

**1. Java Persistence API (JPA)**

* JPA is a Java specification (JSR 338) for accessing, persisting, and managing data between Java objects and relational databases.
* It provides an abstraction layer.
* Does not contain any implementation.
* Requires an implementation like Hibernate.

**2. Hibernate**

* Hibernate is an ORM (Object-Relational Mapping) framework.
* It is the most popular implementation of JPA.
* It manages database communication using Java objects and SQL behind the scenes.

**3. Spring Data JPA**

* Built on top of JPA and Hibernate.
* It’s a Spring module that reduces boilerplate code required by JPA.
* Provides features like automatic query generation and transaction management.
* It doesn't implement JPA but uses a provider (like Hibernate) underneath.

**Code Comparison:**

**Hibernate Example:**

public Integer addEmployee(Employee employee){

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

**Spring Data JPA Example:**

* File name EmployeeRepository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

* File name EmployeeService.java

@Autowired

private EmployeeRepository employeeRepository;

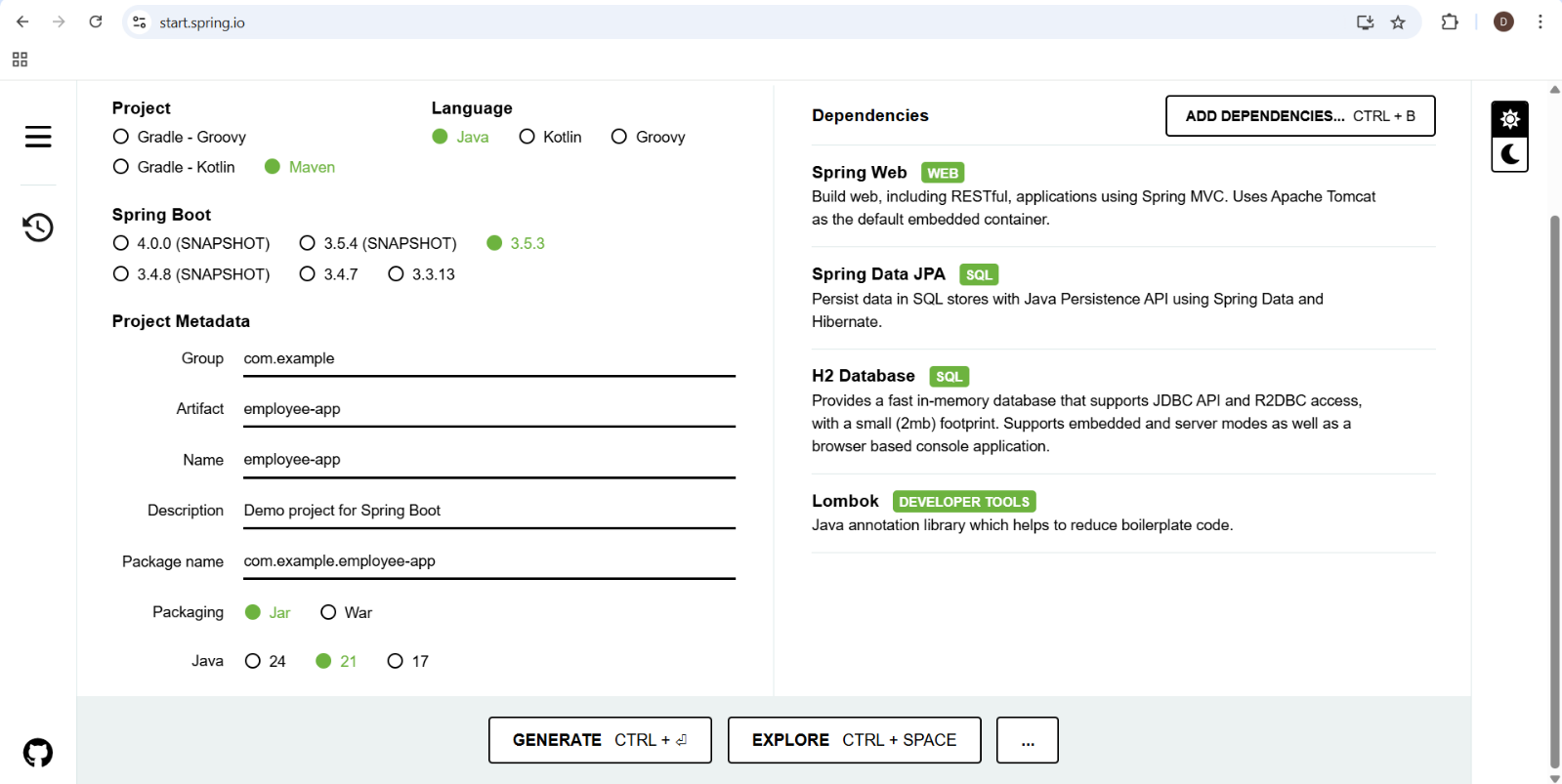
@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

**Implement Spring Boot project using Spring Boot Project:**

****

**application.properties**

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.h2.console.enabled=true

**Entity – Employee.java**

package com.example.employeeapp.entity;

import jakarta.persistence.\*;

import lombok.\*;

@Entity

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

private String name;

private String role;

}

**Repository – EmployeeRepository.java**

package com.example.employeeapp.repository;

import com.example.employeeapp.entity.Employee;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**Service – EmployeeService.java**

package com.example.employeeapp.service;

import com.example.employeeapp.entity.Employee;

import com.example.employeeapp.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public Employee addEmployee(Employee employee) {

return employeeRepository.save(employee);

}

public List<Employee> getAllEmployees() {

return employeeRepository.findAll();

}

}

**Controller – EmployeeController.java**

package com.example.employeeapp.controller;

import com.example.employeeapp.entity.Employee;

import com.example.employeeapp.service.EmployeeService;

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

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

import java.util.List;

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeService employeeService;

@PostMapping

public Employee createEmployee(@RequestBody Employee employee) {

return employeeService.addEmployee(employee);

}

@GetMapping

public List<Employee> getAllEmployees() {

return employeeService.getAllEmployees();

}

}

**Main Class – EmployeeAppApplication.java**

package com.example.employeeapp;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class EmployeeAppApplication {

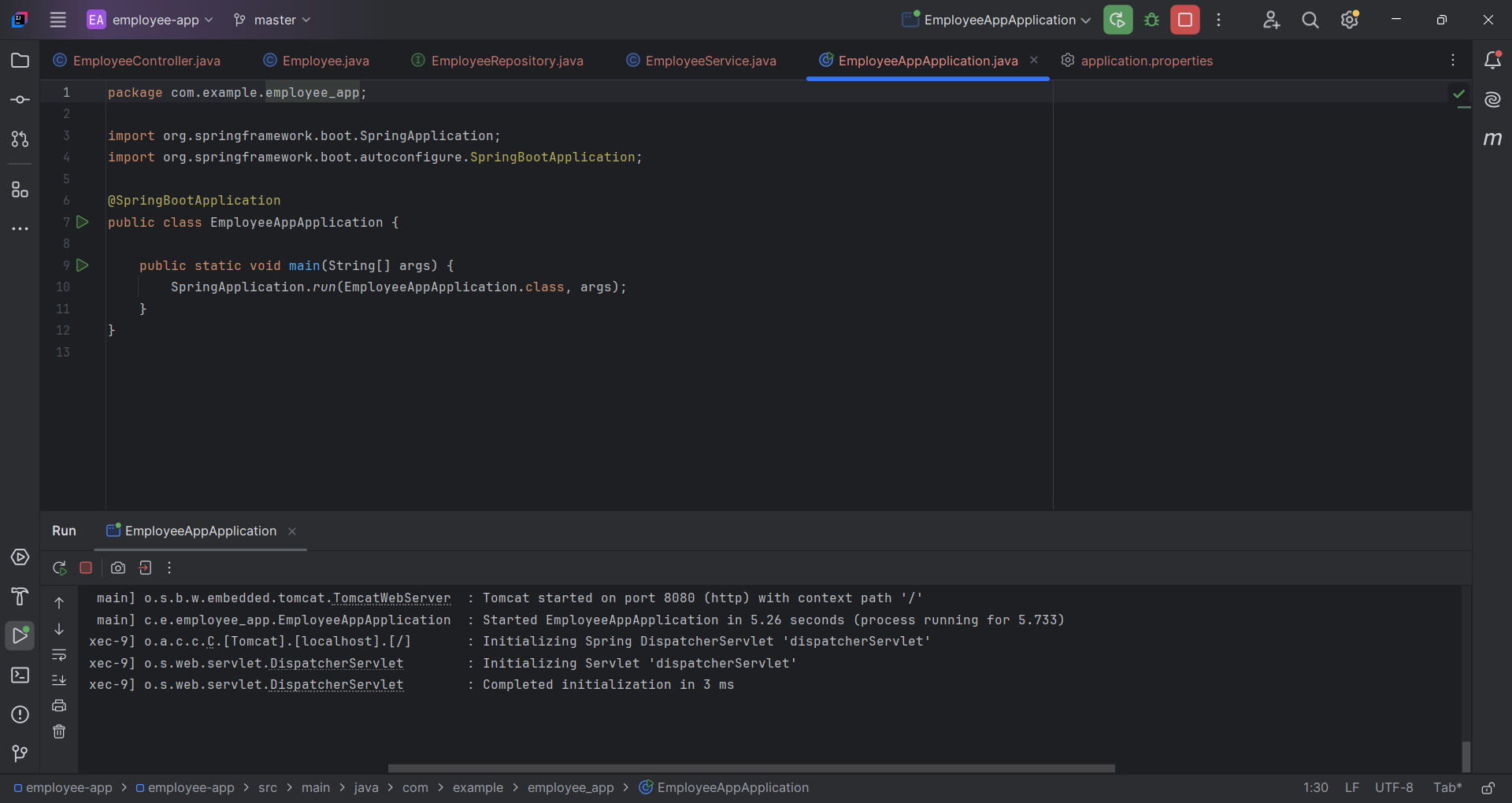
public static void main(String[] args) {

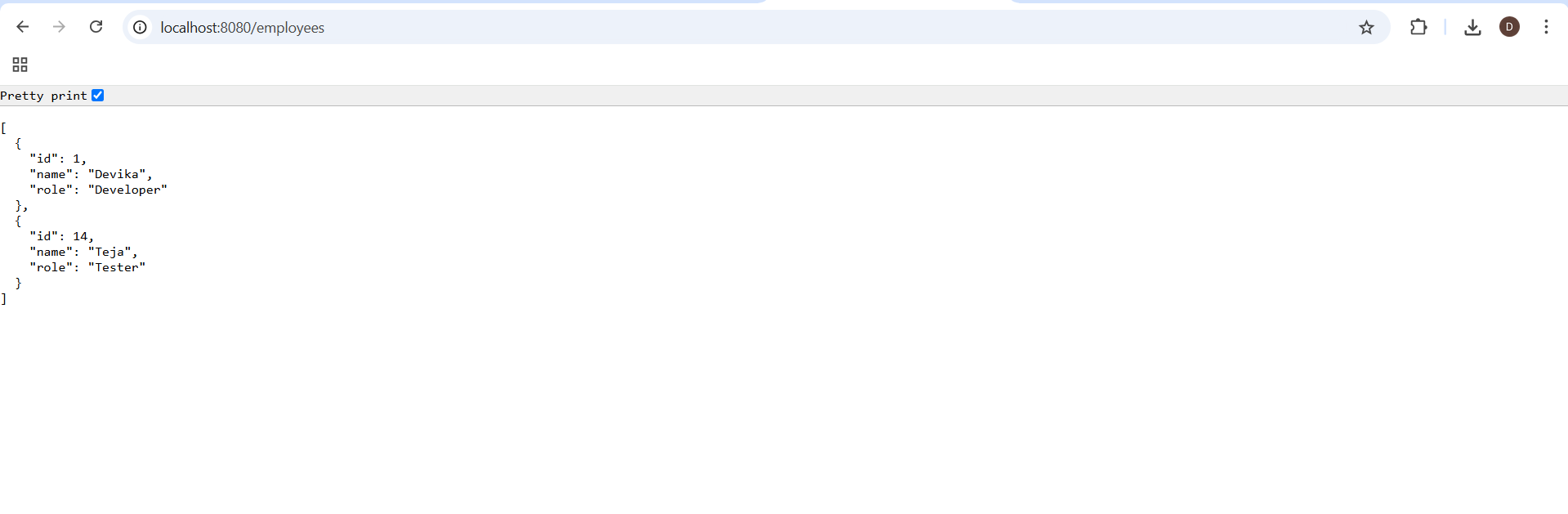
SpringApplication.run(EmployeeAppApplication.class, args);

}

}

**OUTPUT:**

****

****