Digital Nurture 4.0 – Deep Skilling Program

Java Full Stack Engineer (FSE)

Week 3 – Mandatory Exercises Submission

Name: Khushi

GitHub Profile: https://github.com/KhushiSGowda

Email:khushisnagraj12@gmail.com

Skills Covered:

Spring Data JPA with Spring boot, Hibernet

Spring Data JPA – Program and Explanation

**1.Spring Data JPA - Quick Example**

This section demonstrates a basic setup of a Spring Boot project using Spring Data JPA with MySQL.

Steps:

1. Create a Spring Boot project using Spring Initializr with dependencies: Spring Boot DevTools, Spring Data JPA, MySQL Driver.

2. Configure MySQL Database in application.properties.

3. Create 'Country' entity using @Entity and @Table annotations.

4. Create CountryRepository extending JpaRepository.

5. Create CountryService using @Service and @Transactional.

6. Autowire the repository and implement getAllCountries() in service.

7. Test service by invoking it in the main class and logging the results.

## Country.java

@Entity  
@Table(name="country")  
public class Country {  
 @Id  
 @Column(name="code")  
 private String code;  
  
 @Column(name="name")  
 private String name;  
  
 // Getters, Setters, toString()  
}

### CountryRepository.java

@Repository  
public interface CountryRepository extends JpaRepository<Country, String> {  
}

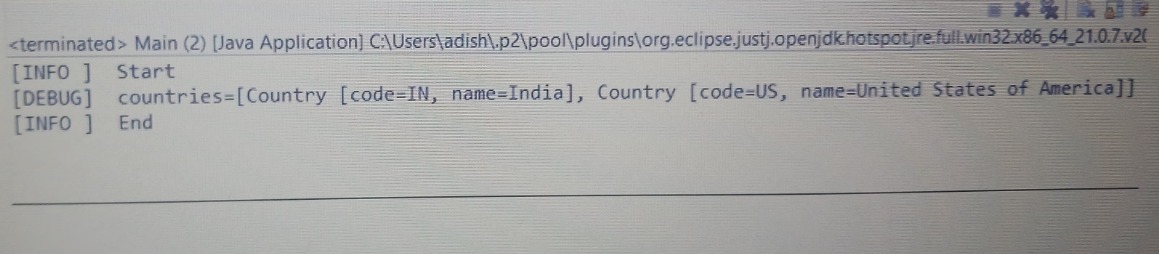
### countryService.java

@Service  
public class CountryService {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
 @Transactional  
 public List<Country> getAllCountries() {  
 return countryRepository.findAll();  
 }  
}

### ormLearnApplication.java

@SpringBootApplication  
public class OrmLearnApplication {  
  
 private static CountryService countryService;  
  
 public static void main(String[] args) {  
 ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);  
 countryService = context.getBean(CountryService.class);  
 testGetAllCountries();  
 }  
  
 private static void testGetAllCountries() {  
 List<Country> countries = countryService.getAllCountries();  
 countries.forEach(System.out::println);  
 }  
}

Output:



**2. Difference between JPA, Hibernate and Spring Data JPA**

* JPA (Java Persistence API):
* - Specification for ORM in Java (JSR 338).
* - Does not provide implementation.
* Hibernate:
* - ORM Tool implementing JPA.
* - Provides native APIs and configuration options.
* Spring Data JPA:
* - Abstraction layer over JPA/Hibernate.
* - Reduces boilerplate code with repository interfaces.
* - Provides query method naming conventions.

## Code Comparison:

### Hibernate:

Session session = factory.openSession();  
Transaction tx = session.beginTransaction();  
session.save(employee);  
tx.commit();  
session.close();

### Spring Data JPA:

@Autowired  
private EmployeeRepository employeeRepository;  
  
@Transactional  
public void addEmployee(Employee employee) {  
 employeeRepository.save(employee);  
}

Output:

