Week 3 – Spring Data JPA with Spring Boot, Hibernate

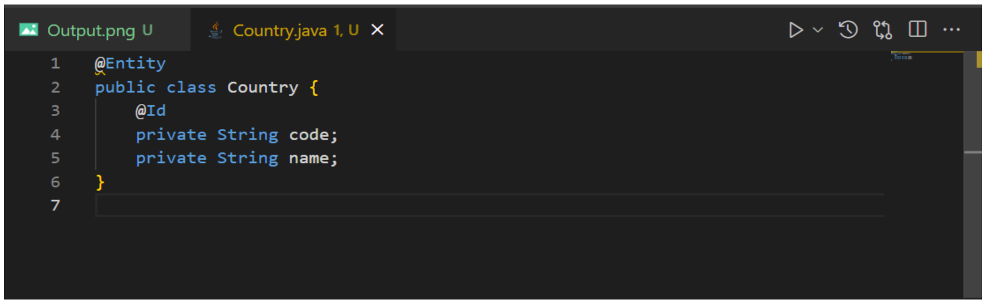
HandsOn 2 - Difference between JPA, Hibernate and Spring Data JPA

1. JPA (Java Persistence API)

- **JPA is a standard specification (JSR 338) that defines how Java objects should interact with relational databases.**
- **It provides a set of annotations and interfaces to map Java objects to database tables using ORM (Object Relational Mapping).**
- **JPA itself does not contain any implementation logic; it acts as a contract or guideline.**

It serves as a blueprint, and tools like Hibernate provide the actual implementation of it.

Example:



```
@Entity
public class Country {
    @Id
    private String code;
    private String name;
}
```

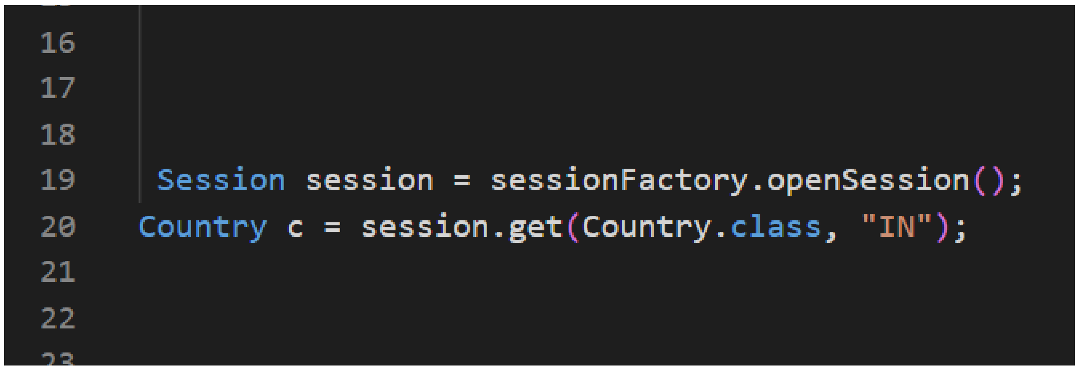
2.Hibernate

- **Hibernate is a widely used implementation of the JPA specification.**
- **It is a fully featured ORM framework that not only supports JPA but also extends it with additional capabilities.**
- **Key features include:**
  - **Caching to improve performance by storing frequently accessed data.**

- **Lazy Loading to load associated data only when required.**
- **Dirty Checking to automatically update only changed data.**
- **HQL (Hibernate Query Language) for writing database queries using object-oriented syntax.**

Hibernate follows JPA standards while offering many more powerful features beyond them.

Example:



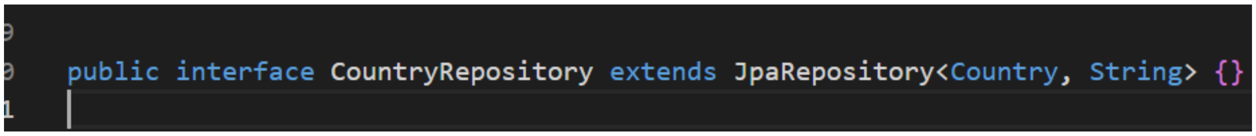
```
 Session session = sessionFactory.openSession();
Country c = session.get(Country.class, "IN");
```

3. Spring Data JPA

- **Spring Data JPA is part of the Spring ecosystem, designed to simplify data access and reduce boilerplate code.**
- **It builds on top of JPA (typically using Hibernate) and auto-generates repository code through method naming conventions.**
- **Developers can write custom queries without writing SQL or HQL in most cases.**

Spring Data JPA is ideal for rapid development and clean data layer integration in Spring-based applications.

Example:



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