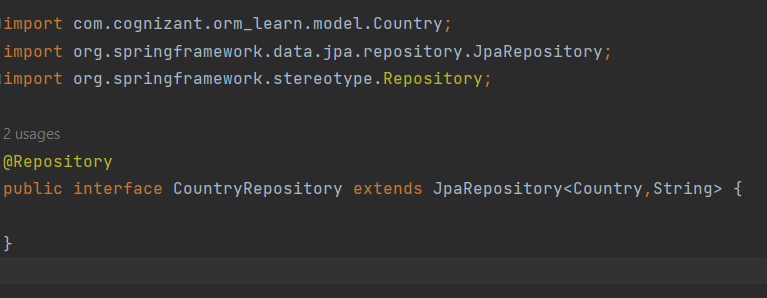
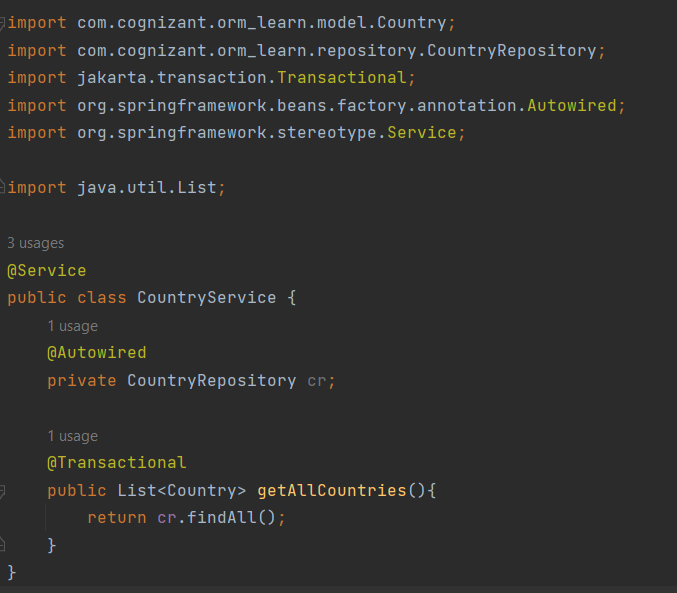
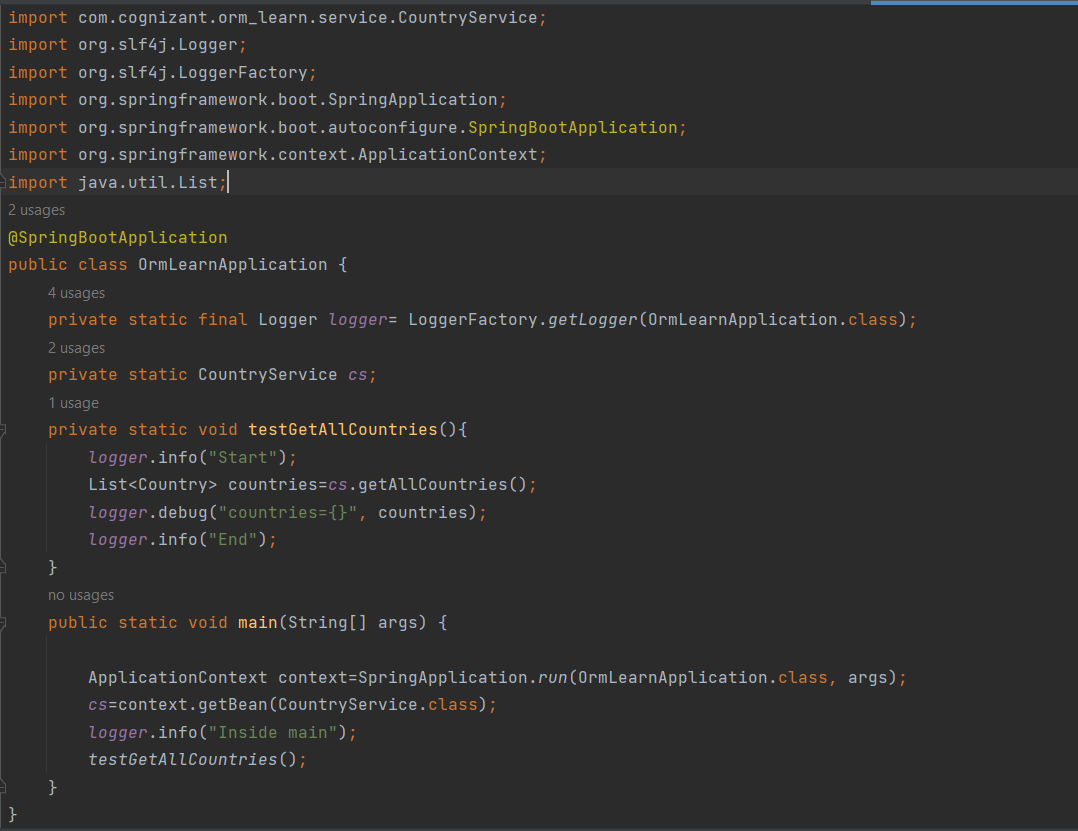
**Hands on 1**

**Spring Data JPA - Quick Example**

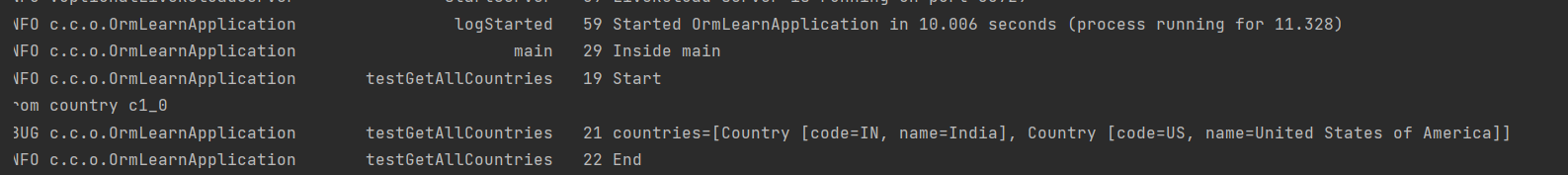








**Output:**



**Hands on 4**

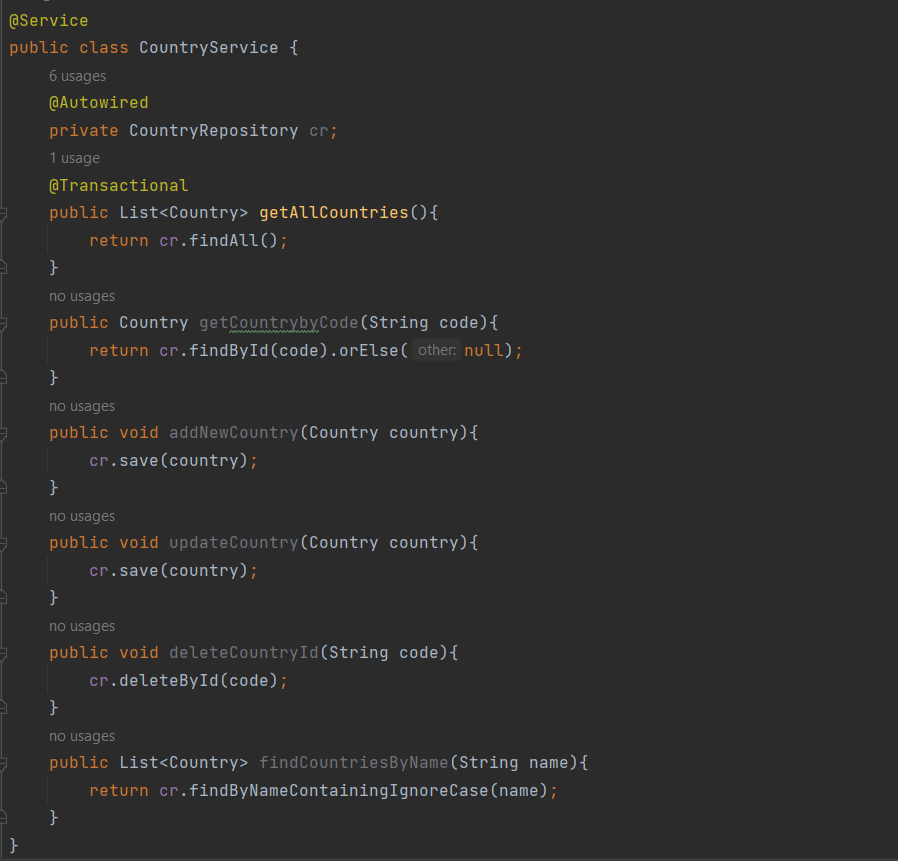
**Difference between JPA, Hibernate and Spring Data JPA**

**Hibernate vs Spring Data JPA**

|  |  |  |
| --- | --- | --- |
| **Feature / Aspect** | **Hibernate** | **Spring Data JPA** |
| **Type** | ORM Framework and JPA implementation | Spring abstraction over JPA (not a JPA provider) |
| **Purpose** | Provides full control over ORM operations using JPA or native APIs | Reduces boilerplate and simplifies data access with repository abstraction |
| **Boilerplate Code** | Requires manual session handling, transaction management, etc. | Very minimal boilerplate using built-in methods like save(), findAll(), etc. |
| **Query Writing** | Uses HQL, Criteria API, or native SQL | Supports derived queries, JPQL, native queries, and custom queries |
| **Integration** | Can be used independently of Spring | Built on top of Spring and integrates seamlessly with Spring Boot |
| **Repository Support** | No built-in repository abstraction — must write DAO classes manually | Provides interfaces like JpaRepository, CrudRepository |
| **Transaction Management** | Manually managed via Hibernate's Transaction object | Managed automatically by Spring with @Transactional annotation |
| **Custom Methods** | Needs to be written explicitly | Supports dynamic query generation using method names |
| **Learning Curve** | Steeper — requires understanding of session, transaction, mappings | Easier for Spring developers — declarative and convention-based |
| **Use Case** | Fine-grained control over ORM operations; legacy or non-Spring applications | Rapid development of CRUD applications in Spring ecosystem |

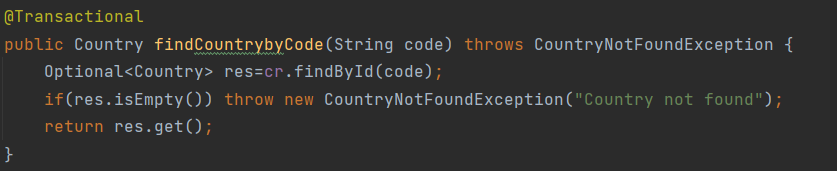
**Hands on 5**

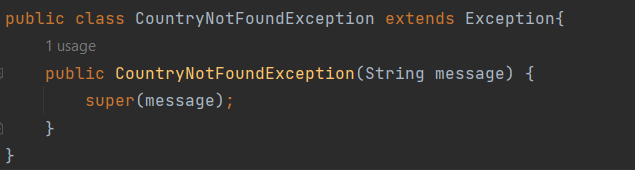
**Implement services for managing Country**



**Hands on 6**

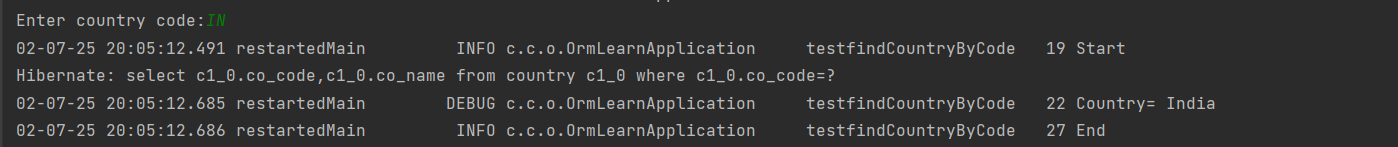
**Find a country based on country code**





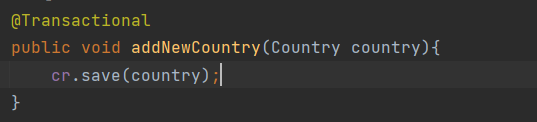


**Output:**



**Hands on 7**

**Add a new country**





**Output:**

