**Spring Data JPA - Quick Example**

1. Created an Eclipse Project using Spring Initilizr

A screenshot of a computer

AI-generated content may be incorrect.

1. Configured application.properties file

A screenshot of a computer

AI-generated content may be incorrect.

1. Created Country Table, and Inserted values in it

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Created Persistence Class – Country.java

A screenshot of a computer

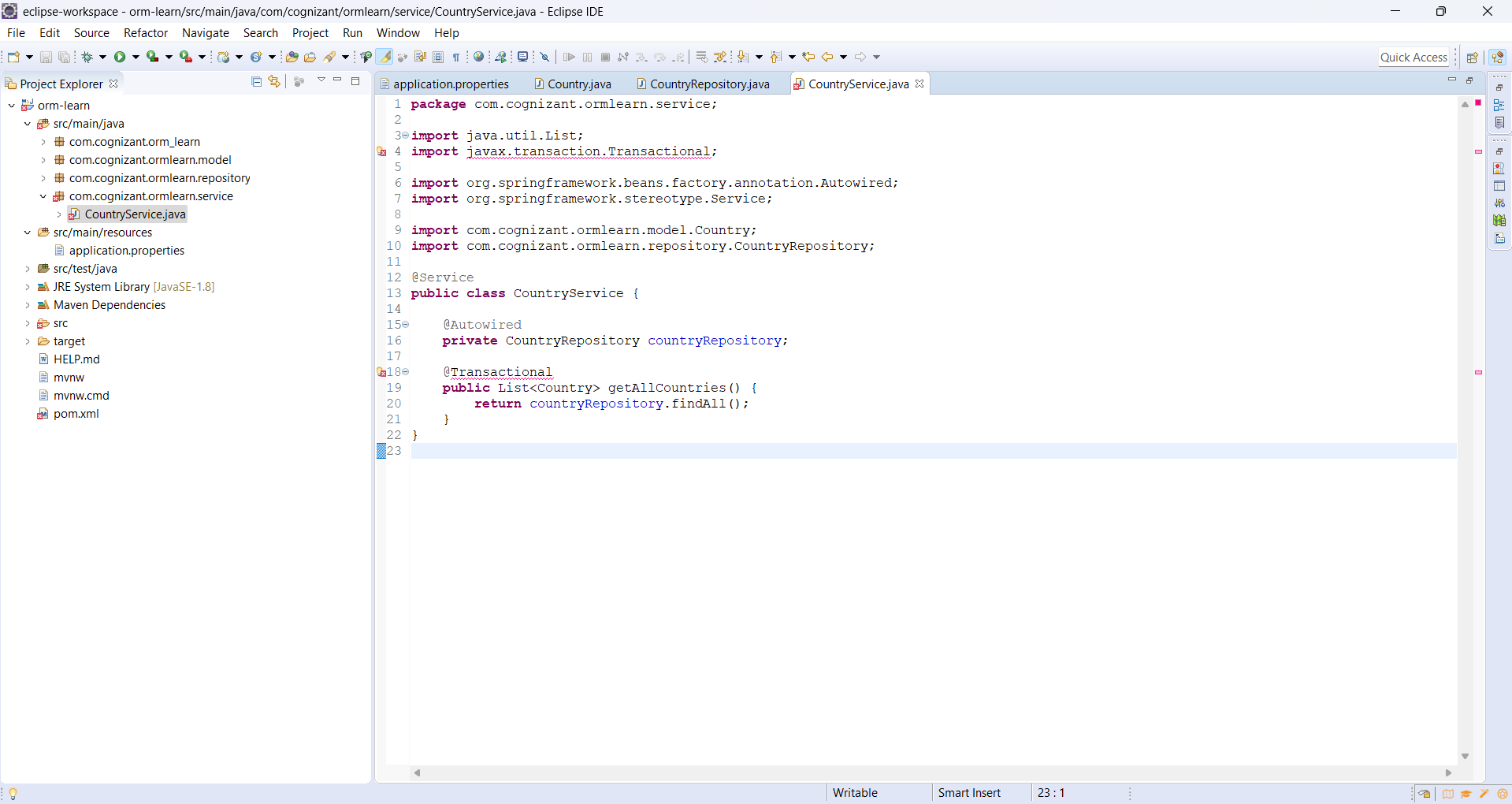
AI-generated content may be incorrect.

Created Repository Class – CountryRepository.java

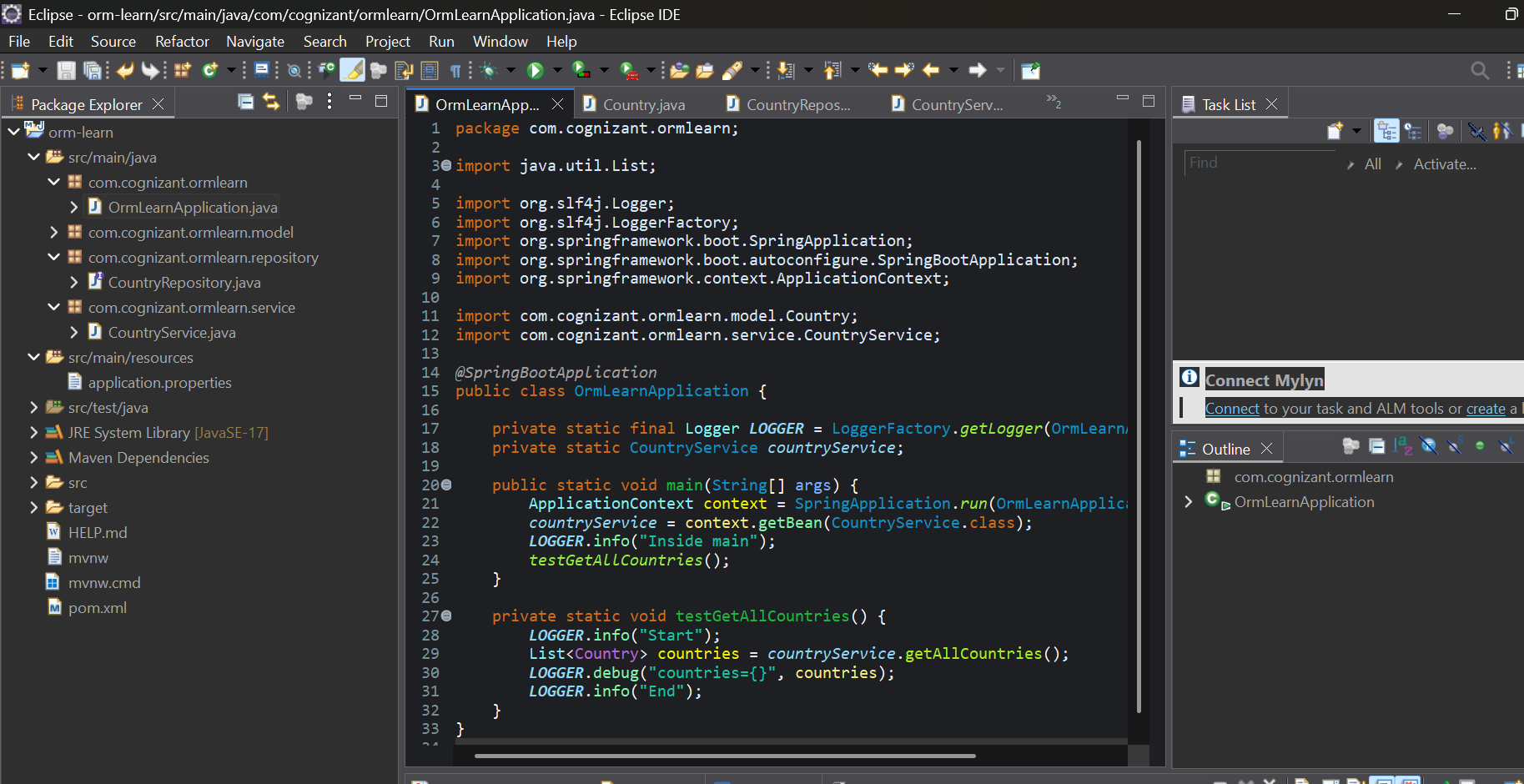
A screenshot of a computer

AI-generated content may be incorrect.

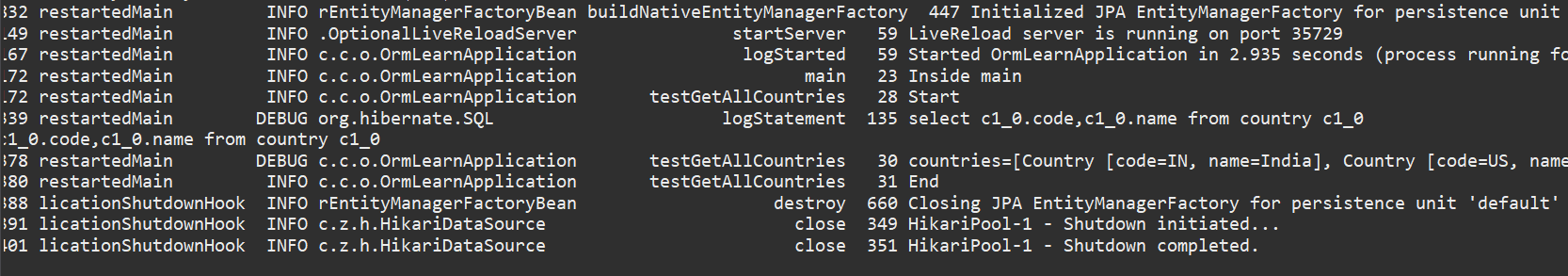
1. Created Service class – CountryService.java



1. Modifying ORM Learn Application Java File



1. It shows correct output,



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

**Hibernate**

* Requires manual setup of session and transaction management.
* Developer needs to:
  + Open a session (factory.openSession()).
  + Begin and commit/rollback transactions manually.
  + Handle exceptions and close the session explicitly.
* More **boilerplate code** is needed to perform a simple database operation like saving an entity.

**Spring Data JPA**

* Built on top of JPA and abstracts away most of the boilerplate.
* Developer only needs to:
  + Create a repository interface extending JpaRepository.
  + Autowire the repository and call the save() method.
* Spring handles:
  + Session management.
  + Transaction management (using @Transactional).
  + Exception handling internally.
* Leads to **cleaner, more concise, and readable** code with **minimal configuration**.