**Hands-on 7: Add a New Country**

This hands-on exercise guides you through implementing functionality to add a new country to the database using Spring Data JPA.

**✅ Step-by-Step Implementation**

**1. Add a New Method in CountryService**

**Method Signature:**

@Transactional

public void addCountry(Country country);

**Implementation:**

@Transactional

public void addCountry(Country country) {

countryRepository.save(country);

}

**2. Add a Test Method in OrmLearnApplication**

**Method:**

private static void testAddCountry() {

LOGGER.info("Start");

Country newCountry = new Country();

newCountry.setCode("XY");

newCountry.setName("Xyland");

countryService.addCountry(newCountry);

try {

Country addedCountry = countryService.findCountryByCode("XY");

LOGGER.debug("Added Country: {}", addedCountry);

} catch (CountryNotFoundException e) {

LOGGER.error("Country not found: {}", e.getMessage());

}

LOGGER.info("End");

}

**3. Modify the main() Method in OrmLearnApplication.java**

Add the new test method call:

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

countryService = context.getBean(CountryService.class);

testGetAllCountries(); // Optional: existing test

testFindCountryByCode(); // Optional: previous test

testAddCountry(); // ✅ NEW: Add country test

}

**✅ Result Verification**

After running the application, verify the following:

* Console should log:
* Start
* Added Country: Country [code=XY, name=Xyland]
* End
* You can also **validate using MySQL Workbench**:
* SELECT \* FROM country WHERE co\_code = 'XY';

**📌 Notes**

* The @Transactional annotation ensures that the save() operation is committed as a database transaction.
* Spring automatically manages session and transaction boundaries for you.
* The save() method of JpaRepository is capable of both inserting new records and updating existing ones based on the primary key.