Spring Data JPA Hands-On Solutions

# Hands-On 1: Spring Data JPA Quick Example

This hands-on demonstrates setting up a Spring Boot project using Spring Data JPA and MySQL.

## application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn  
spring.datasource.username=root  
spring.datasource.password=root  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
  
spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect  
  
logging.level.org.hibernate.SQL=trace  
logging.level.org.hibernate.type.descriptor.sql=trace

## Country Table SQL

CREATE TABLE country (  
 co\_code VARCHAR(2) PRIMARY KEY,  
 co\_name VARCHAR(50)  
);  
  
INSERT INTO country VALUES ('IN', 'India');  
INSERT INTO country VALUES ('US', 'United States of America');

## Country.java

@Entity  
@Table(name = "country")  
public class Country {  
 @Id  
 @Column(name = "co\_code")  
 private String code;  
  
 @Column(name = "co\_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 final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);  
 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() {  
 LOGGER.info("Start");  
 List<Country> countries = countryService.getAllCountries();  
 LOGGER.debug("countries={}", countries);  
 LOGGER.info("End");  
 }  
}

# Hands-On 5: CRUD Features using CountryService

## Find Country by Code

@Transactional  
public Country findCountryByCode(String code) throws CountryNotFoundException {  
 return countryRepository.findById(code).orElseThrow(() ->  
 new CountryNotFoundException("Country not found for code: " + code));  
}

## Add New Country

@Transactional  
public void addCountry(Country country) {  
 countryRepository.save(country);  
}

## Update Country

@Transactional  
public void updateCountry(String code, String name) {  
 Country country = countryRepository.findById(code)  
 .orElseThrow(() -> new CountryNotFoundException("Country not found"));  
 country.setName(name);  
 countryRepository.save(country);  
}

## Delete Country

@Transactional  
public void deleteCountry(String code) {  
 countryRepository.deleteById(code);  
}