**Spring Data JPA - Hands-on 1**

**Demo project for Spring Data JPA and Hibernate**

**application.properties**

**Code:**

# Spring Framework and application log  
logging.level.org.springframework=info  
logging.level.com.cognizant=debug  
  
# Hibernate logs for displaying executed SQL, input and output  
logging.level.org.hibernate.SQL=trace  
logging.level.org.hibernate.type.descriptor.sql=trace  
  
# Log pattern  
logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n  
  
# Database configuration  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn  
spring.datasource.username=root  
spring.datasource.password=root  
  
# Hibernate configuration  
spring.jpa.hibernate.ddl-auto=validate  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

**OrmLearnApplication.java**

**Code:**

@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);  
 LOGGER.info("Inside main");  
 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");  
 }  
}

**Country.java**

**Code:**

@Entity  
@Table(name="country")  
public class Country {  
 @Id  
 @Column(name="code")  
 private String code;  
  
 @Column(name="name")  
 private String name;  
  
 // Getters, Setters and toString()  
}

**CountryRepository.java**

**Code:**

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

**CountryService.java**

**Code:**  
@Service  
public class CountryService {  
  
 @Autowired  
 private CountryRepository countryRepository;  
  
 @Transactional  
 public List<Country> getAllCountries() {  
 return countryRepository.findAll();  
 }  
}

**MySQL Table & Insert**

**Code:**  
create schema ormlearn;  
use ormlearn;  
  
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');

**Output:**

**Code:**  
Inside main  
Start  
countries=[Country [code=IN, name=India], Country [code=US, name=United States of America]]  
End