Instructions

The goal of this exercise is to create a simple RESTful service using Spring MVC and Spring Boot.

1) Start out by creating a new project in STS using File -> New -> "Spring Starter Project"

2) In the "Dependencies" screen, select Web -> "Spring Web" as well as SQL -> "Spring Data JPA" and SQL -> "MySQL Driver" (or "MS SQL Driver")

3) Verify that the following dependencies have been added to the generated POM file:

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <scope>runtime</scope>
</dependency>
```

4) Create a simple JPA entity. Select a table from a pre-popultated DB such as the "sakila" or "Adventure Works" schema in MySQL and map it to a JPA entity. Note: You don't have to include all the columns

5) Create the data-access layer for your entity using Spring Data interface JpaRepository. Here is an example:

```
@Repository

public interface MyEntityRepository extends JpaRepository<MyEntity, MyEntityIdType> {

}
```

Hint: This will auto-magically create the data-access layer for you.

6) Create a REST Controller using the @RestController annotation

7) Create a sample controller method annotated with @RequestMapping

8) @Autowire the repository in your controller class

9) Inside the controller method, call one of the methods of your repository type and return the result.

10) Test your application by running the "main". You should be able to view the result of your RESTful service at: http://localhost:8080/ (or any relative URLs from the roots, depending on how you map your @RequestMapping)

11) Connect your app to a real database and produce some results!

Hint: You can use the following configuration for configuring the DB connection. If you are using MS SQL Server, then please refer to the simple examples that I have shared this morning.

```
server:
  port: 8081
```

```yaml
spring:
  application:
    name: sample-springdata-service
  datasource:
    url: jdbc:mysql://localhost/sakila
    username: sakila
    password: sakila
    driverClassName: com.mysql.cj.jdbc.Driver
  jpa:
    show-sql: true
    hibernate:
      naming:
        physical-strategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
        implicit-strategy: org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl
    properties:
      hibernate:
        dialect: org.hibernate.dialect.MySQL8Dialect
```