Web API Design with Spring Boot Week 3 Coding Assignment

Points possible: 70

| Category | Criteria | % of Grade |
|---------------|---|------------|
| Functionality | Does the code work? | 25 |
| Organization | Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are | 25 |
| Creativity | concise and clear. Student solved the problems | 25 |
| Cicativity | presented in the assignment using creativity and out of the box thinking. | 23 |
| Completeness | All requirements of the assignment are complete. | 25 |

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Here's a friendly tip: as you watch the videos, code along with the videos. This will help you with the homework. When a screenshot is required, look for the icon: You will keep adding to this project throughout this part of the course. When it comes time for the final project, use this project as a starter.

Project Resources: https://github.com/promineotech/Spring-Boot-Course-Student-Resources

Coding Steps:

- In the application you've been building add a DAO layer:
 - Add the package, com.promineotech.jeep.dao.
 - In the new package, create an interface named JeepSalesDao.

- In the same package, create a class named DefaultJeepSalesDao that implements JeepSalesDao.
- Add a method in the DAO interface and implementation that returns a list of Jeep models (class Jeep) and takes the model and trim parameters. Here is the method signature:

List<Jeep> fetchJeeps(JeepModel model, String trim);

- In the Jeep sales service implementation class, inject the DAO interface as an instance variable. The instance variable should be private and should be named jeepSalesDao. Call the DAO method from the service method and store the returned value in a local variable named jeeps. Return the value in the jeeps variable (we will add to this later).
- In the DAO implementation class (DefaultJeepSalesDao):
 - Add the class-level annotation: @Service.
 - Add a log statement in DefaultJeepSalesDao.fetchJeeps() that logs the model and trim level. Run the integration test. Produce a screenshot showing the DAO implementation class and the log line in the IDE's console.
 - In DefaultJeepSalesDao, inject an instance variable of type NamedParameterJdbcTemplate.
 - Write SQL to return a list of Jeep models based on the parameters: model and trim. Be sure to utilize the SQL Injection prevention mechanism of the NamedParameterJdbcTemplate using :model_id and :trim_level in the query.
 - Add the parameters to a parameter map as shown in the video. Don't forget to convert the JeepModel enum value to a String (i.e., params.put("model_id", model.toString());)
 - Call the query method on the NamedParameterJdbcTemplate instance variable to return a list of Jeep model objects. Use a RowMapper to map each row of the result set. Remember to convert modelId to a JeepModel. See the video for details. Produce a screenshot to show the complete method in the implementation class.
- Add a getter in the Jeep class for modelPK. Add the @JsonIgnore annotation to the getter to exclude the modelPK value from the returned object.
- Run the test to produce a green status bar. Produce a screenshot showing the test and the green status bar.

Screenshots of Code:

```
DefaultJeepSalesService.java
                            JeepSalesDao.java
                                                 DefaultJeepSalesDao.java X
J Fetc
 package com.promineotech.jeep.dao;
  3@ import java.math.BigDecimal;
 4 import java.sql.ResultSet;
 5 import java.sql.SQLException;
 6 import java.util.HashMap;
 7 import java.util.List;
 8 import java.util.Map;
 9 import org.springframework.beans.factory.annotation.Autowired;
 10 import org.springframework.jdbc.core.RowMapper;
 import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
12 import org.springframework.stereotype.Component;
13 import org.springframework.stereotype.Service;
14 import com.promineotech.jeep.entity.Jeep;
15 import com.promineotech.jeep.entity.JeepModel;
16 import lombok.extern.slf4j.Slf4j;
18 @Component
19 @Service
20 @SLf4j
21 public class DefaultJeepSalesDao implements JeepSalesDao {
230
    @Autowired
      private NamedParameterJdbcTemplate jdbcTemplate;
      @Override
 260
27
      public List<Jeep> fetchJeeps(JeepModel model, String trim) {
        Log.debug("DAO: model={}, trim={}", model, trim);
        String sql = ""
            + "WHERE model id = :model id AND trim level = :trim level";
        Map<String, Object> params = new HashMap<>();
        params.put("model_id", model.toString());
        params.put("trim level", trim);
        return jdbcTemplate.query(sql, params,
            new RowMapper<>() {
450
              @Override
              public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
46
                return Jeep.builder()
                     .basePrice(new BigDecimal(rs.getString("base_price")))
                     .modelId(JeepModel.valueOf(rs.getString("model_id")))
                     .modelPK(rs.getLong("model_pk"))
                     .numDoors(rs.getInt("num_doors"))
                     .trimLevel(rs.getString("trim_level"))
                     .wheelSize(rs.getInt("wheel size"))
                     .build();
               // formatter:on
              }});
      }
```

Screenshots of Running Application:

```
DefaultJeepSalesService.java
                                                   JeepSalesDao.java
                                                                                      🕡 *DefaultJeepSalesDao.java 🗶 🕡 FetchJeepTes
   10 import org.springtramework.stereotype.component;
        import org.springframework.stereotype.Service:
import com.promineotech.jeep.sales/src/main/java/com/promineotech/jeep/service/DefaultJeepSalesService/pep.service/pep.service/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesService/DefaultJeepSalesSer
        import com.promineotech.jeep.entity.JeepModel;
         import lombok.extern.slf4j.Slf4j;
   16 @Component
         @Service
   18 @Slf4j
         public class DefaultJeepSalesDao implements JeepSalesDao {
   210
            @Autowired
 22
             private NamedParameterJdbcTemplate jbdcTemplate;
   240
             @Override
 25
             public List<Jeep> fetchJeeps(JeepModel model, String trim) {
                Log.debug("DAO: model={}, trim={}", model, trim);
A29
                String sql = "'
                       + "WHERE model id = :model id AND trim level = :trim level";
                Map<String, Object> params = new HashMap<>();
                params.put("model_id", model);
                params.put("trim_level", trim);
 ×41
                 return jdbcTemplate.query(sql, params, new RowMapper<>() {});
          <
 🦹 Problems 🏿 🛭 Javadoc 🔼 Declaration 📮 Console 🗶
terminated> FetchJeepTest [JUnit] C:\Users\Sojian\STS\sts-4.11.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot
11:44:08.540 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiatin
11:44:08.560 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiatin
11:44:08.600 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating
11:44:08.612 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBoot
11:44:08.616 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader
11:44:08.616 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader
11:44:08.617 [main] INFO org.springframework.test.context.support.AbstractContextLoader
11:44:08.617 [main] INFO org.springframework.test.context.support.AnnotationConfigContex
11:44:08.762 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandid
11:44:08.763 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBoot
11:44:08.884 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBoo
11:44:08.884 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBoot
11:44:08.905 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBoot
11:44:08.909 [main] DEBUG org.springframework.test.context.support.AbstractDirtiesContex
11:44:08.922 [main] DEBUG org.springframework.test.context.support.DependencyInjectionTe
11:44:08.955 [main] DEBUG org.springframework.core.env.StandardEnvironment - Activating
11:44:08.959 [main] DEBUG org.springframework.test.context.support.TestPropertySourceUti
                                                             (v2.5.2)
  :: Spring Boot ::
 2021-07-09 11:44:09.381 INFO 11348 --- [
                                                                                            main] c.p.jeep.controller.FetchJeepT
```

```
📱 Package Explorer 🚽 🗗 JUnit 🗶
                                                                    DefaultJeepSalesService.java
                                                                                                                                                  DefaultJeepSalesDao.ia
                                                                                                                 J JeepSalesDao, java
                                                                      package com.promineotech.jeep.controller;
        Finished after 5.684 seconds
                                                                      3● import static org.assertj.core.api.Assertions.assertThat;
4 import java.util.List;
  Runs: 1/1
                     Errors: 0
                                         * Failures: 0
                                                                          import org.junit.jupiter.api.Test;
// import org.springframework.beans.factory.annotation.Autowire
                                                                          import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.boot.test.context.SpringBootTest.WebB
 > 🌆 FetchJeepTest [Runner: JUnit 5] (0.989 s)
                                                                          import org.springframework.core.ParameterizedTypeReference;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
                                                                                ort org.springframework.http.ResponseEntity;
                                                                          // import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.test.context.ActiveProfiles;
import org.springframework.test.context.jdbc.Sql;
import org.springframework.test.context.jdbc.Sql;
                                                                          import org.springframework.test.context.jdbc.SqlConfig;
// import org.springframework.test.jdbc.JdbcTestUtils;
                                                                     import com.promineotech.jeep.controller.support.FetchJeepTestSup
import com.promineotech.jeep.entity.Jeep;
import com.promineotech.jeep.entity.JeepModel;
                                                                     23 @SpringBootTest(webEnvironment = WebEnvironment.RANDOM PORT)
                                                                     25 @ActiveProfiles("test")
                                                                          @Sql(scripts = {
                                                                                 "classpath:flyway/migrations/V1.0_Jeep_Schema.sql",
"classpath:flyway/migrations/V1.1_Jeep_Data.sql"},
config = @SqlConfig(encoding = "utf-8"))
                                                                     35€
                                                  圆 严 誓
 Failure Trace
                                                                                int numcows = JdbcTestUtils.countRowsInTable(jdbcTemplate, System.out.println("num=" + numcows);
                                                                     43
44
                                                                     460
                                                                                // Given: a valid model, trim and URI
JeepModel model = JeepModel.WRANGLER;
                                                                                 String trim = "Sport";
String uri = String.format("%s?model=%s&trim=%s", getBaseUri
                                                                     53
54
                                                                                 ResponseEntity<List<Jeep>> response = getRestTemplate().exch
                                                                     55
56
57
58
59
60
                                                                                 assertThat(response.getStatusCode()).isEqualTo(HttpStatus.Ok
                                                                                 List<Jeep> actual = response.getBody();
List<Jeep> expected = buildExpected();
                                                                                 assertThat(actual).isEqualTo(expected);
 ♥ Boot Dashboard X
                                                                     65 }
     □ ∅ □ □ ✓ / □ ♥ Y +
```

URL to GitHub Repository:

https://github.com/sojians/Week-15-Spring