


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 concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
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:

- 1) In the application you've been building add a DAO layer:
 - a) Add the package, `com.promineotech.jeepp.dao`.
 - b) In the new package, create an interface named `JeepSalesDao`.
 - c) In the same package, create a class named `DefaultJeepSalesDao` that implements `JeepSalesDao`.

- d) 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);
```

- 2) 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).
- 3) In the DAO implementation class (`DefaultJeepSalesDao`):
 - a) Add the class-level annotation: `@Service`.
 - b) 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. 
 - c) In `DefaultJeepSalesDao`, inject an instance variable of type `NamedParameterJdbcTemplate`.
 - d) 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.
 - e) 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());`)
 - f) 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. 
- 4) 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.
- 5) Run the test to produce a green status bar. Produce a screenshot showing the test and the green status bar. 

Screenshots of Code:

```
DefaultJeepSalesController.java | FetchJeepTest.java | JeepSalesController.java | JeepSalesDao.java | *DefaultJeepSalesDao.java x
1 package com.promineotech.jeeo.dao;
2
3 import java.util.List;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Component;
6 import com.promineotech.jeeo.entity.Jeeo;
7 import com.promineotech.jeeo.entity.JeeoModel;
8 import lombok.extern.slf4j.Slf4j;
9
10 @Component
11 @Slf4j
12 public class DefaultJeepSalesDao implements JeepSalesDao {
13
14     @Autowired
15     private NamedParameterJdbcTemplate jdbcTemplate;
16
17     @Override
18     public List<Jeeo> fetchJeeps(JeeoModel model, String trim) {
19         log.debug("DAO: model={}, trim={}", model, trim);
20         return null;
21     }
22
23 }
24
```

Outline

- com.promineotech.jeeo.dao
 - DefaultJeepSalesDao
 - log - Logger
 - jdbcTemplate - NamedParameterJdbcTemplate
 - fetchJeeps(JeeoModel, String) : List<Jeeo>

Problems | Javadoc | Declaration | Console

<terminated> FetchJeepTest [JUnit] /Applications/SpringToolSuite4.app/Contents/Eclipse/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64_17.0.1.v20211116-1657/jre/bin/java (Mar 21, 2022, 7:55:31 PM - 7:55:45 PM)

19:55:36.218 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Neither @ContextConfiguration nor @ContextHierarchy found for test class [com.promineotech.jeeo.FetchJeepTest]

19:55:36.227 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.promineotech.jeeo.controller.FetchJeepTest]

19:55:36.228 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.promineotech.jeeo.controller.FetchJeepTest]

19:55:36.228 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.promineotech.jeeo.controller.FetchJeepTest]

19:55:36.231 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.promineotech.jeeo.controller.FetchJeepTest]

19:55:36.582 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate component class: file [/Users/eduardosupler/Desktop/BackEndCoding/JeepSales/src/main/java/com/promineotech/jeeo/controller/DefaultJeepSalesController.class]

19:55:36.583 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootTestConfiguration com.promineotech.jeeo.JeeoSales for test class com.promineotech.jeeo.FetchJeepTest

19:55:36.652 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - @TestExecutionListeners is not present for class [com.promineotech.jeeo.controller.FetchJeepTest]

19:55:36.653 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestExecutionListener, org.springframework.test.context.support.MethodProfiler, org.springframework.test.context.support.DirtiesContextTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DefaultTestExecutionListener, org.springframework.test.context.support.MethodProfiler, org.springframework.test.context.support.DirtiesContextTestExecutionListener]

19:55:36.687 [main] DEBUG org.springframework.test.context.support.AbstractDirtiesContextTestExecutionListener - Before test class: context [DefaultTestContext@2a7686a7 testClass = FetchJeepTest]

19:55:36.784 [main] DEBUG org.springframework.test.context.support.DependencyInjectionTestExecutionListener - Performing dependency injection for test context [DefaultTestContext@2a7686a7 testClass = FetchJeepTest]

Spring Boot :: (v2.6.4)

2022-03-21 19:55:37.311 INFO 42800 --- [main] c.p.jeeo.controller.FetchJeepTest : Starting FetchJeepTest using Java 17.0.1 on Carlys-MBP.lan with PID 42800 (started by eduardosupler)

2022-03-21 19:55:37.313 DEBUG 42800 --- [main] c.p.jeeo.controller.FetchJeepTest : Running with Spring Boot v2.6.4, Spring v5.3.16

2022-03-21 19:55:37.314 INFO 42800 --- [main] c.p.jeeo.controller.FetchJeepTest : The following 1 profile is active: "test"

2022-03-21 19:55:43.361 INFO 42800 --- [main] c.p.jeeo.controller.FetchJeepTest : Started FetchJeepTest in 6.61 seconds (JVM running for 9.277)

2022-03-21 19:55:45.082 INFO 42800 --- [o-auto-1-exec-1] c.p.jeeo.DefaultJeepSalesController : model=WRANGLER, trim=Sport

2022-03-21 19:55:45.084 INFO 42800 --- [o-auto-1-exec-1] c.p.jeeo.service.DefaultJeepSalesService : The fetchJeeps method was called with model=WRANGLER and trim=Sport

2022-03-21 19:55:45.084 DEBUG 42800 --- [o-auto-1-exec-1] c.p.jeeo.dao.DefaultJeepSalesDao : DAO: model= WRANGLER, trim=Sport

```
DefaultJeepSalesController.java  FetchJeepTest.java  JeepSalesController.java  JeepSalesDao.java  DefaultJeepSalesDao.java  Jeep.java

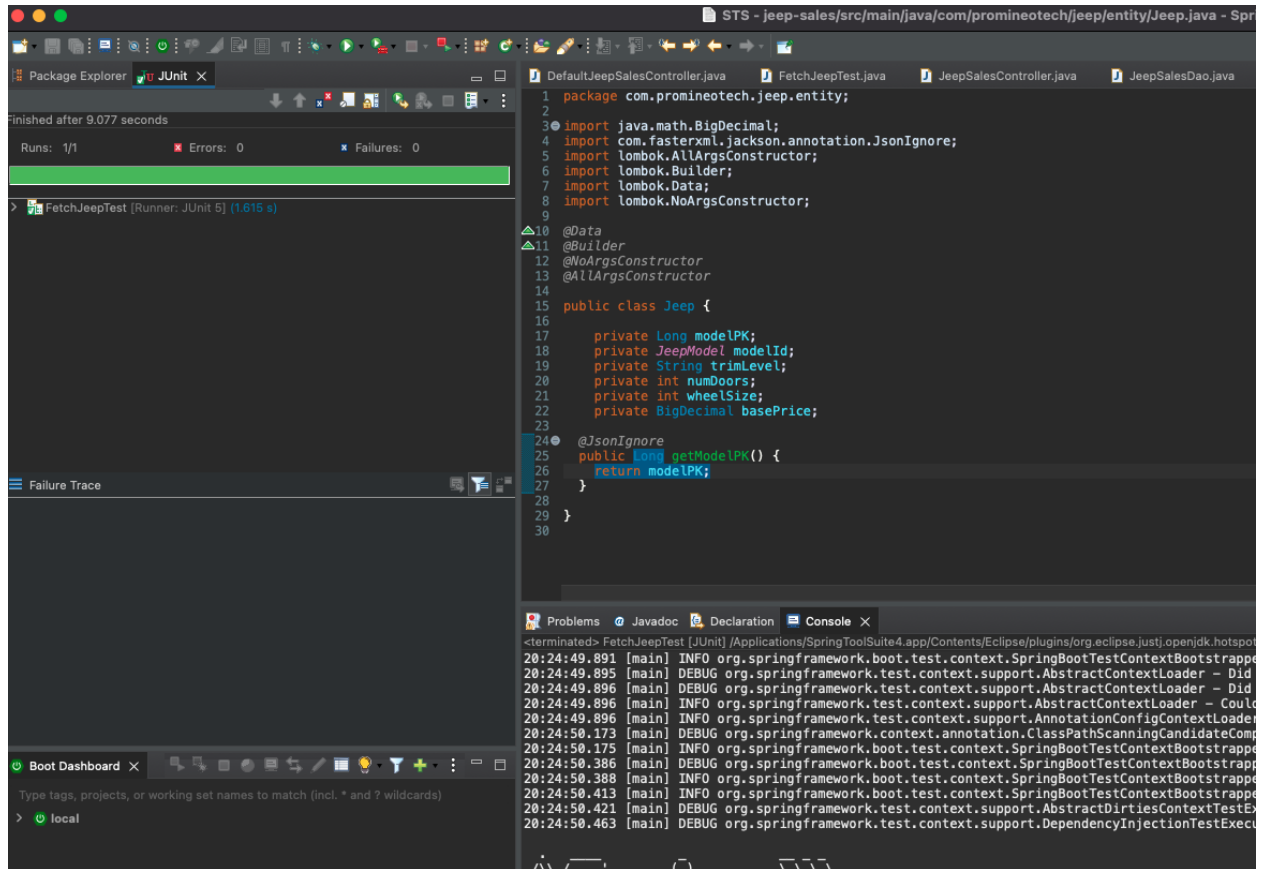
24 @Autowired
25 private NamedParameterJdbcTemplate jdbcTemplate;
26
27 @Override
28 public List<Jeep> fetchJeeps(JeepModel model, String trim) {
29     log.debug("DAO: model={}, trim={}", model, trim);
30
31     String sql = ""
32         + "SELECT * "
33         + "FROM models "
34         + "Where model_id = :model_id AND trim_level = :trim_level";
35
36     Map<String, Object> params = new HashMap<>();
37     params.put("model_id", model.toString());
38     params.put("trim_level", trim);
39
40     return jdbcTemplate.query(sql, params,
41         new RowMapper<>() {
42             public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
43                 return Jeep.builder()
44                     .basePrice(new BigDecimal(rs.getString("base_price")))
45                     .modelId(JeepModel.valueOf(rs.getString("model_id")))
46                     .modelPK(rs.getLong("model_pk"))
47                     .numDoors(rs.getInt("num_doors"))
48                     .trimLevel(rs.getString("trim_level"))
49                     .wheelSize(rs.getInt("wheel_size"))
50                     .build();
51             }
52         });
53 }
54
55 }
56

Problems  Javadoc  Declaration  Console X

<terminated> FetchJeepTest [JUnit] /Applications/SpringToolSuite4.app/Contents/Eclipse/plugins/org.eclipse.justi.openjdk.hotspot.jre.full.macosx.x86_64.j70.1.v20211116-1657/jre/bin/java. (Mar 21, 2022, 8:22:15 PM - 8:
20:22:19.894 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Neither @ContextConfiguration nor @ContextHierarchy found for
20:22:19.902 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.promineot
20:22:19.903 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.promineot
20:22:19.903 [main] INFO org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for test class [com.promineot
20:22:19.903 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.promineot
20:22:20.207 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate component class: file [/Users/
20:22:20.209 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.promineotech.jeep.JeepSalesController
20:22:20.414 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - @TestExecutionListeners is not present for class [com.promineotech.jeep.JeepSalesController]
20:22:20.415 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Loaded default TestExecutionListener class names from location [classpath:org/springframework/boot/test/context/executionlisteners.properties]
20:22:20.440 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.junit4.SpringJUnit4ClassRunner, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtiesContextTestExecutionListener, org.springframework.test.context.support.SpringJUnit4ClassRunner$SpringMethodExecutionListener]
20:22:20.448 [main] DEBUG org.springframework.test.context.support.DependencyInjectionTestExecutionListener - Before test class: context [DefaultTestContext@20:22:20.448]
20:22:20.475 [main] DEBUG org.springframework.test.context.support.DependencyInjectionTestExecutionListener - Performing dependency injection for test context [DefaultTestContext@20:22:20.475]

Spring Boot (v2.6.4)

2022-03-21 20:22:21.235 INFO 42919 --- [main] c.p.jeep.controller.FetchJeepTest : Starting FetchJeepTest using Java 17.0.1 on Carlys-MBP.lan
2022-03-21 20:22:21.237 DEBUG 42919 --- [main] c.p.jeep.controller.FetchJeepTest : Running with Spring Boot v2.6.4, Spring v5.3.16
2022-03-21 20:22:21.237 INFO 42919 --- [main] c.p.jeep.controller.FetchJeepTest : The following 1 profile is active: "test"
2022-03-21 20:22:26.573 INFO 42919 --- [main] c.p.jeep.controller.FetchJeepTest : Started FetchJeepTest in 6.015 seconds (JVM running for 8.6
2022-03-21 20:22:27.680 INFO 42919 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : The fetchJeeps method was called with model=WRANGLER, trim=Sport
```



Screenshots of Running Application:

The screenshot shows an IDE with several Java files open: `DefaultJeepSalesController.java`, `FetchJeepTest.java`, `JeepSalesController.java`, `JeepSalesDao.java`, and `*DefaultJeepSalesDao.java`. The `DefaultJeepSalesDao.java` file is the active editor, showing a class that implements `JeepSalesDao`. It has a private field `jdbcTemplate` and an `@Override` method `public List<Jeep> fetchJeeps(JeepModel model, String trim) { ... }`. The console at the bottom shows the output of a Spring Boot application, including the Spring logo and various log messages. The logs indicate that the application is running on a Mac OS X machine, and the `FetchJeepTest` class is being executed. The logs also show that the `fetchJeeps` method was called with `model=WRANGLER` and `trim=Sport`.

```
1 package com.promineotech.jeep.dao;
2
3 import java.util.List;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Component;
6 import com.promineotech.jeep.entity.Jeep;
7 import com.promineotech.jeep.entity.JeepModel;
8 import lombok.extern.slf4j.Slf4j;
9
10 @Component
11 @Slf4j
12 public class DefaultJeepSalesDao implements JeepSalesDao {
13
14     @Autowired
15     private NamedParameterJdbcTemplate jdbcTemplate;
16
17     @Override
18     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
19         log.debug("DAO: model={}, trim={}", model, trim);
20         return null;
21     }
22
23 }
24
```

Spring Boot (v2.6.4)

```
2022-03-21 19:55:37.311 INFO 42800 --- [main] c.p.jeep.controller.FetchJeepTest : Starting FetchJeepTest using Java 17.0.1 on Carlys-MBP.lan with PID 42800 (started by edwards)
2022-03-21 19:55:37.313 DEBUG 42800 --- [main] c.p.jeep.controller.FetchJeepTest : Running with Spring Boot v2.6.4, Spring v5.3.16
2022-03-21 19:55:37.314 INFO 42800 --- [main] c.p.jeep.controller.FetchJeepTest : The following 1 profile is active: "test"
2022-03-21 19:55:43.361 INFO 42800 --- [main] c.p.jeep.controller.FetchJeepTest : Started FetchJeepTest in 6.61 seconds (JVM running for 9.277)
2022-03-21 19:55:45.082 INFO 42800 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2022-03-21 19:55:45.084 INFO 42800 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : The fetchJeeps method was called with model=WRANGLER and trim=Sport
2022-03-21 19:55:45.084 DEBUG 42800 --- [o-auto-1-exec-1] c.p.jeep.dao.DefaultJeepSalesDao : DAO: model= WRANGLER, trim=Sport
```

URL to GitHub Repository:

<https://github.com/edsupler/SpringHW>