

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:

```
FetchJeepTest.java | DefaultJeepSalesDao.java | DefaultJeepSalesService.java | JeepSalesService.java | DefaultJeepSalesController.java | JeepModel.java | Jeep.java
25 @Override
26 public List<Jeep> fetchJeeps(JeepModel model, String trim) {
27     Log.info("DAO: model={}, trim={}", model, trim);
28
29     // @formatter:off
30     String sql = "
31         + "SELECT * "
32         + "FROM models "
33         + "WHERE model_id = :model_id "
34         + "AND trim_level = :trim_level";
35     // @formatter:on
36
37     Map<String, Object> params = new HashMap<>();
38     params.put("model_id", model.toString());
39     params.put("trim_level", trim);
40
41     return jdbcTemplate.query(sql, params, new RowMapper<>() {
42
43     @Override
44     public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
45
46         // @formatter:off
47         return Jeep.builder()
48             .basePrice(new BigDecimal(rs.getString("base_price")))
49             .modelId(JeepModel.valueOf(rs.getString("model_id")))
50             .modelPK(rs.getLong("model_pk"))
51             .numDoors(rs.getInt("num_doors"))
52             .trimLevel(rs.getString("trim_level"))
53             .wheelSize(rs.getInt("wheel_size"))
54             .build();
55         // @formatter
56     }
57 }
```

Screenshots of Running Application:

The screenshot shows an IDE with a test run failure. The test is `testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied()` in `FetchJeepTest`. The failure trace indicates an `org.opentest4j.AssertionFailedError` with the message: "expected: Jeep(modelPK=null, modelId=WRANGLER, trimLevel=Sport) but was: Jeep(modelPK=1, modelId=WRANGLER, trimLevel=Sport)". The console log shows the application running successfully, with the following output:

```
<terminated> FetchJeepTest [JUnit] C:\Users\soha\sts-4.15.3.RELEASE\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (Oct 8, 2022, 11:41:03 AM) [pid: 36224]
2022-10-08 11:41:12.551 INFO 36224 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 m
2022-10-08 11:41:12.586 INFO 36224 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2022-10-08 11:41:12.586 INFO 36224 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : These fetchJeeps Method was cal
2022-10-08 11:41:12.587 INFO 36224 --- [o-auto-1-exec-1] c.p.jeep.dao.DefaultJeepSalesDao : DAO: model=WRANGLER, trim=Sport
2022-10-08 11:41:12.796 INFO 36224 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiat
2022-10-08 11:41:12.801 INFO 36224 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown complet
```

spring-boot-jeeep-sales/src/main/java/com/promineotech/jeeep/dao/DefaultJeepSalesDao.java - Spring Tool Suite 4

Finished after 5.428 seconds
Runs: 1/1 Errors: 0 Failures: 1

FetchJeepTest (Runner: JUnit 5) (0.905 s)
testThatJeepsAreReturnedWhenValidModelAndTrim

```

25
26
27 @Override
28 public List<Jeep> fetchJeeps(JeepModel model, String trim) {
29     Log.info("DAO: model={}, trim={}", model, trim);
30
31     // @formatter:off
32     String sql = "
33         + "SELECT * "
34         + "FROM models "
35         + "WHERE model_id = :model_id "
36         + "AND trim_level = :trim_level";
37     // @formatter:on
38
39     Map<String, Object> params = new HashMap<>();
40     params.put("model_id", model.toString());
41     params.put("trim_level", trim);
42
43     return jdbcTemplate.query(sql, params, new RowMapper<>() {

```

Failure Trace

```

org.opentest4j.AssertionFailedError:
expected:
[Jeep(modelPK=null, modelId=WRANGLER, trimLevel=Sport)]
but was:
[Jeep(modelPK=1, modelId=WRANGLER, trimLevel=Sport)]

```

Problems Javadoc Declaration Console

```

2022-10-08 11:45:18.516 INFO 32688 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed
2022-10-08 11:45:19.938 INFO 32688 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-10-08 11:45:20.145 INFO 32688 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-10-08 11:45:20.387 INFO 32688 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 5282
2022-10-08 11:45:20.398 INFO 32688 --- [main] c.p.jeeep.controller.FetchJeepTest : Started FetchJeepTest in 3.984 s
2022-10-08 11:45:21.057 INFO 32688 --- [o-auto-1-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet
2022-10-08 11:45:21.057 INFO 32688 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-10-08 11:45:21.059 INFO 32688 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
2022-10-08 11:45:21.112 INFO 32688 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2022-10-08 11:45:21.112 INFO 32688 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : These fetchJeeps Method was called
2022-10-08 11:45:21.112 INFO 32688 --- [o-auto-1-exec-1] c.p.jeeep.dao.DefaultJeepSalesDao : DAO: model=WRANGLER, trim=Sport
2022-10-08 11:45:21.348 INFO 32688 --- [onShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated
2022-10-08 11:45:21.353 INFO 32688 --- [onShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed

```

spring-boot-jeeep-sales/src/main/java/com/promineotech/jeeep/dao/DefaultJeepSalesDao.java - Spring Tool Suite 4

Finished after 5.39 seconds
Runs: 1/1 Errors: 0 Failures: 0

FetchJeepTest (Runner: JUnit 5) (0.998 s)

```

71 list.add(Jeep.builder()
72     .modelId(JeepModel.WRANGLER)
73     .trimLevel("Sport")
74     .numDoors(2)
75     .wheelSize(17)
76     .basePrice(new BigDecimal("28475.00"))
77     .build()
78 );
79
80 list.add(Jeep.builder()
81     .modelId(JeepModel.WRANGLER)
82     .trimLevel("Sport")
83     .numDoors(4)
84     .wheelSize(17)
85     .basePrice(new BigDecimal("31975.00"))
86     .build()
87 );
88
89 // @formatter:on

```

Problems Javadoc Declaration Console

```

2022-10-08 11:49:00.686 INFO 13432 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-10-08 11:49:00.893 INFO 13432 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-10-08 11:49:01.129 INFO 13432 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 5282
2022-10-08 11:49:01.140 INFO 13432 --- [main] c.p.jeeep.controller.FetchJeepTest : Started FetchJeepTest in 3.78 s
2022-10-08 11:49:01.895 INFO 13432 --- [o-auto-1-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet
2022-10-08 11:49:01.895 INFO 13432 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-10-08 11:49:01.896 INFO 13432 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2022-10-08 11:49:01.929 INFO 13432 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2022-10-08 11:49:01.930 INFO 13432 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : These fetchJeeps Method was called
2022-10-08 11:49:01.930 INFO 13432 --- [o-auto-1-exec-1] c.p.jeeep.dao.DefaultJeepSalesDao : DAO: model=WRANGLER, trim=Sport
2022-10-08 11:49:02.183 INFO 13432 --- [onShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated
2022-10-08 11:49:02.185 INFO 13432 --- [onShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed

```

URL to GitHub Repository:

<https://github.com/sohailjaan2002/Week15-Coding-Assignment>