

Web API Design with Spring Boot Week 15 Coding Assignment

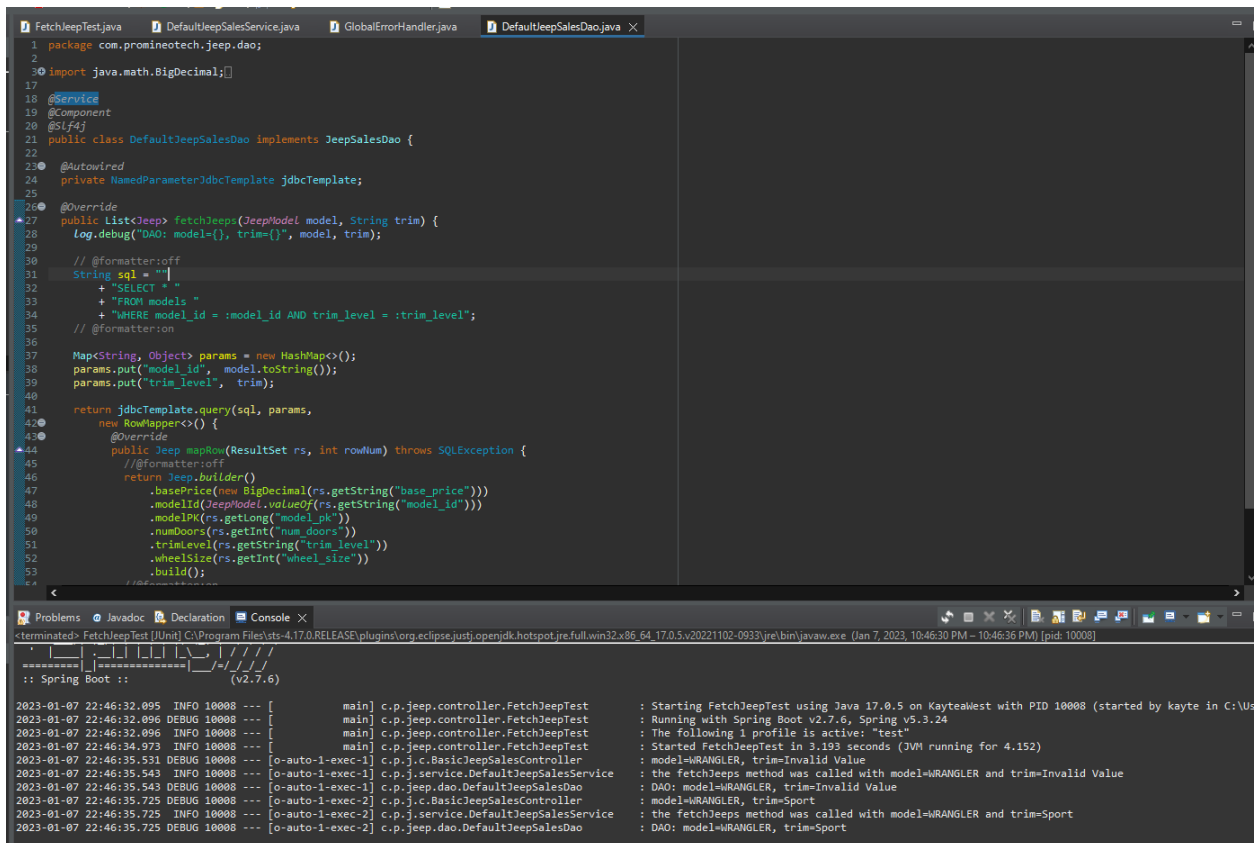
Points possible: 75

URL to GitHub Repository: <https://github.com/kayteawest/PromineoSpringBoot.git>

URL to Public Link of your Video:

https://drive.google.com/file/d/1VeN4P-I-9wuiu5-W98kA_VLjMYpWKYEh/view?usp=sharing

Screen Shots:



```
1 package com.promineotech.jeep.dao;
2
3 import java.math.BigDecimal;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 @Service
19 @Component
20 @Slf4j
21 public class DefaultJeepSalesDao implements JeepSalesDao {
22
23     @Autowired
24     private NamedParameterJdbcTemplate jdbcTemplate;
25
26
27     @Override
28     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
29         log.debug("DAO: model={}, trim={}", model, trim);
30
31         // @formatter:off
32         String sql = "
33             + "SELECT *
34             + "FROM models
35             + "WHERE model_id = :model_id AND trim_level = :trim_level";
36         // @formatter:on
37
38         Map<String, Object> params = new HashMap<>();
39         params.put("model_id", model.toString());
40         params.put("trim_level", trim);
41
42         return jdbcTemplate.query(sql, params,
43             new RowMapper<>() {
44                 @Override
45                 public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
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                 }
56             });
57     }
58 }
```

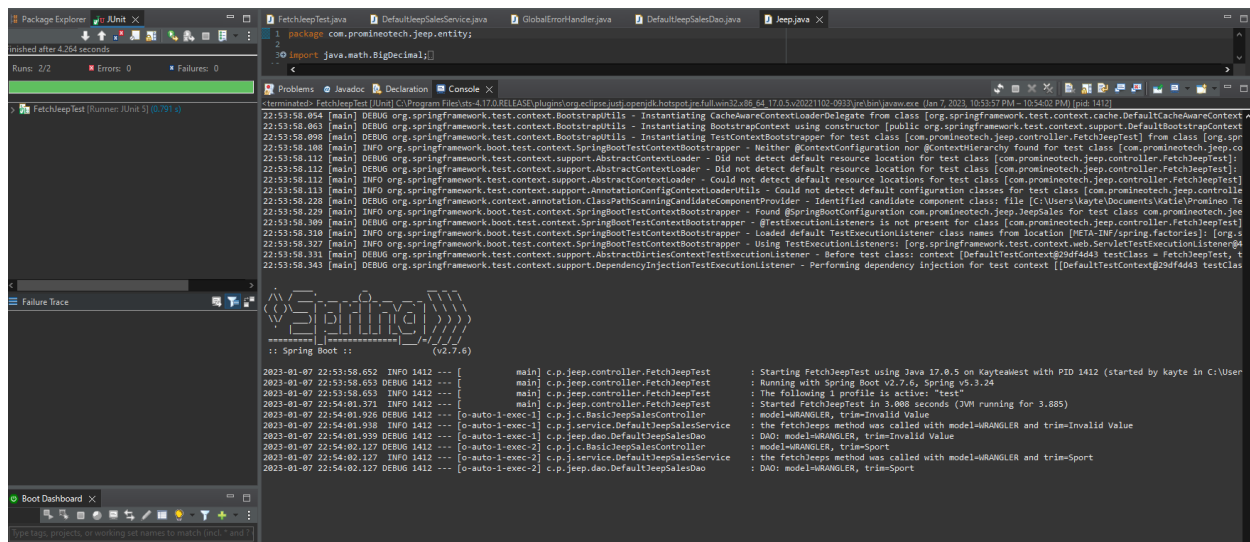
```

=====
:: Spring Boot ::
(v2.7.6)

2023-01-07 22:46:32.095 INFO 10008 --- [main] c.p.jeep.controller.FetchJeepTest : Starting FetchJeepTest using Java 17.0.5 on Kayteawest with PID 10008 (started by kayte in C:\Uo
2023-01-07 22:46:32.096 DEBUG 10008 --- [main] c.p.jeep.controller.FetchJeepTest : Running with Spring Boot v2.7.6, Spring v5.3.24
2023-01-07 22:46:32.096 INFO 10008 --- [main] c.p.jeep.controller.FetchJeepTest : The following 1 profile is active: "test"
2023-01-07 22:46:34.973 INFO 10008 --- [main] c.p.jeep.controller.FetchJeepTest : Started FetchJeepTest in 3.193 seconds (JVM running for 4.152)
2023-01-07 22:46:35.531 DEBUG 10008 --- [o-auto-1-exec-1] c.p.j.c.BasicJeepSalesController : model=WRAangler, trim=Invalid Value
2023-01-07 22:46:35.543 INFO 10008 --- [o-auto-1-exec-1] c.p.j.s.DefaultJeepSalesService : the fetchJeeps method was called with model=WRAangler and trim=Invalid Value
2023-01-07 22:46:35.543 DEBUG 10008 --- [o-auto-1-exec-1] c.p.j.d.DefaultJeepSalesDao : DAO: model=WRAangler, trim=Invalid Value
2023-01-07 22:46:35.725 DEBUG 10008 --- [o-auto-1-exec-2] c.p.j.c.BasicJeepSalesController : model=WRAangler, trim=Sport
2023-01-07 22:46:35.725 INFO 10008 --- [o-auto-1-exec-2] c.p.j.s.DefaultJeepSalesService : the fetchJeeps method was called with model=WRAangler and trim=Sport
2023-01-07 22:46:35.725 DEBUG 10008 --- [o-auto-1-exec-2] c.p.j.d.DefaultJeepSalesDao : DAO: model=WRAangler, trim=Sport
```

Web API Design with Spring Boot Week 15 Coding Assignment


```
FetchJeepTest.java | DefaultJeepSalesService.java | GlobalErrorHandler.java | DefaultJeepSalesDao.java X
19 @Component
20 @Slf4j
21 public class DefaultJeepSalesDao implements JeepSalesDao {
22
23     @Autowired
24     private NamedParameterJdbcTemplate jdbcTemplate;
25
26     @Override
27     public List<Jeep> fetchJeeps(JeepModel model, String trim) {
28         log.debug("DAO: model={}, trim={}", model, trim);
29
30         // @formatter:off
31         String sql = ""
32             + "SELECT *"
33             + "FROM models "
34             + "WHERE model_id = :model_id 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,
42             new RowMapper<>() {
43                 @Override
44                 public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
45                     // @formatter:off
46                     return Jeep.builder()
47                         .basePrice(new BigDecimal(rs.getString("base_price")))
48                         .modelId(JeepModel.valueOf(rs.getString("model_id")))
49                         .modelPK(rs.getLong("model_pk"))
50                         .numDoors(rs.getInt("num_doors"))
51                         .trimLevel(rs.getString("trim_level"))
52                         .wheelSize(rs.getInt("wheel_size"))
53                         .build();
54                     // @formatter:on
55                 }
56             });
57     }
58 }
```




Web API Design with Spring Boot Week 15 Coding Assignment

Instructions :

1. Follow the **Coding Steps** below to complete this assignment.

- In Spring Tool Suite (STS), or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed.
- Use your existing repo or create a new repository on GitHub for this week's assignment and push your completed code to the repo, including your entire Maven Project Directory (e.g., jeep-sales) and any additional files (e.g. .sql files) that you create. In addition, screenshot your ERD and push the screenshot to your GitHub repo.
- Include the screenshots into this Assignment Document indicated by: 
- Create a video showcasing your work:
 - In this video: record and present your project verbally while showing the results of the working project.
 - Easy way to Create a video: Start a meeting in Zoom, share your screen, open Eclipse with the code and your Console window, start recording & record yourself describing and running the program showing the results.
 - Your video should be a maximum of 5 minutes.
 - Upload your video with a public link.
 - Easy way to Create a Public Video Link: Upload your video recording to YouTube with a public link.


2. In addition, please include the following in your Coding Assignment Document:

- The requested screenshots, indicated by: 
- The URL for this week's GitHub repository.
- The URL of the public link of your video.

3. Save the Coding Assignment Document as a .pdf and do the following:

- Push the .pdf to the GitHub repo for this week.
- Upload the .pdf to the LMS in your Coding Assignment Submission.

Web API Design with Spring Boot Week 15 Coding Assignment


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.jeep.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`.

Web API Design with Spring Boot Week 15 Coding Assignment

- 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. 