





Web API Design with Spring Boot Week 15 Coding Assignment

URL to GitHub Repository: <https://github.com/SamKnight823/Springboot-Project>

URL to Public Link of your Video: <https://youtu.be/qA9PoT1TTW4>

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 functionality into your Video when you see: 
- Create a video showcasing your work:
 - In this video: record and present your project verbally while showing the results of the working project. Don't forget to include the requested functionality, indicated by: 
 - 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 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 you should include something in your video submission, look for the icon: 

Note: 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).



Web API Design with Spring Boot Week 15 Coding Assignment

- 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. In your video, show the DAO implementation class and the log line in the IDE's console.

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with `com.promineotech.jee.dao` and `DefaultJeepSalesDao`.
- DefaultJeepSalesDao.java:** The main file being edited. It contains the following code:

```
1 package com.promineotech.jee.dao;
2
3 import java.util.List;
4 import org.springframework.stereotype.Service;
5 import com.promineotech.jee.entity.Jee;
6 import com.promineotech.jee.entity.JeeModel;
7 import lombok.extern.slf4j.Slf4j;
8
9 @Service
10 @Slf4j
11 public class DefaultJeepSalesDao implements JeepSalesDao {
12
13
14
15     public List<Jee> fetchJeeps(JeeModel model, String trim) {
16         log.debug("DAO: model={}, trim={}, model, trim);
17         return null;
18     }
19 }
20 }
```
- Console:** Shows the output of the test run, including the log statement:

```
2023-04-11 15:05:37.757 INFO 16572 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-04-11 15:05:37.759 INFO 16572 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
2023-04-11 15:05:37.811 INFO 16572 --- [o-auto-1-exec-1] c.p.j.e.DefaultJeepSalesController : The fetchJeeps method was called with mo
2023-04-11 15:05:37.811 INFO 16572 --- [o-auto-1-exec-1] c.p.j.e.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2023-04-11 15:05:37.866 INFO 16572 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2023-04-11 15:05:37.968 INFO 16572 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
```

- 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.



PROMINEO TECH

Web API Design with Spring Boot Week 15 Coding Assignment

Remember to convert `modelId` to a `JeepModel`. See the video for details. In your video, show the complete method in the implementation class. 📺

```
21 @Autowired
22 private NamedParameterJdbcTemplate jdbcTemplate;
23
24 public List<Jeep> fetchJeeps(JeepModel model, String trim) {
25     log.debug("DAO: model={}, trim={}, model, trim");
26
27     //formatter:off
28     String sql = ""
29         + "SELECT * "
30         + "FROM models "
31         + "WHERE model_id = :model_id AND trim_level = :trim_level";
32     //formatter:on
33
34     Map<String, Object> params = new HashMap<String, Object>();
35     params.put("model_id", model.toString());
36     params.put("trim_level", trim);
37
38     return jdbcTemplate.query(sql, params,
39         new RowMapper<Jeep>() {
40             @Override
41             public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
42                 //formatter:off
43                 return Jeep.builder()
44                     .basePrice(new BigDecimal(rs.getString("base_price")))
45                     .modelId(rs.getModelId().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                 //formatter:on
52             }
53         })
54 }
```

2023-04-11 14:58:49.549 INFO 22784 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2023-04-11 14:58:49.588 INFO 22784 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2023-04-11 14:58:49.588 INFO 22784 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : The fetchJeeps method was called with mo
2023-04-11 14:58:49.628 INFO 22784 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2023-04-11 14:58:49.631 INFO 22784 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.

- 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. In your video, show the test and the green status bar. 📺



PROMINEO TECH

Web API Design with Spring Boot Week 15 Coding Assignment

The screenshot displays an IDE window for a Spring Boot application. The main editor shows a Java file named `FetchJeepTest.java` with the following code:

```
51 List<Jeep> expected = buildExpected();
52 assertEquals(response.getBody(), expected);
53 System.out.println(response.getBody());
54 }
55
56 protected List<Jeep> buildExpected() {
57     List<Jeep> list = new LinkedList<Jeep>();
58
59     // @formatter:off
60     list.add(Jeep.builder()
61         .modelId(JeepModel.WRANGLER)
62         .trimLevel("Sport")
63         .numDoors(2)
64         .wheelSize(17)
65         .basePrice(new BigDecimal("28475.00"))
66         .build());
67
68     list.add(Jeep.builder()
69         .modelId(JeepModel.WRANGLER)
70         .trimLevel("Sport")
71         .numDoors(4)
72         .wheelSize(17)
73         .basePrice(new BigDecimal("31975.00"))
74         .build());
75
76     // @formatter:on
77     return list;
78 }
79
80 }
```

The left sidebar shows the Package Explorer with the project structure. The bottom panel displays the Console output, which includes the following log messages:

```
2023-04-11 14:58:49.549 INFO 22784 --- [o-auto-1-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms
2023-04-11 14:58:49.588 INFO 22784 --- [o-auto-1-exec-1] c.p.j.c.DefaultJeepSalesController : model=WRANGLER, trim=Sport
2023-04-11 14:58:49.588 INFO 22784 --- [o-auto-1-exec-1] c.p.j.service.DefaultJeepSalesService : The fetchJeeps method was called with mo
1Jeep(modelId=null, modelId=WRANGLER, trimLevel=Sport, numDoors=2, wheelSize=17, basePrice=28475.00), Jeep(modelId=null, modelId=WRANGLER, tr
2023-04-11 14:58:49.828 INFO 22784 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...
2023-04-11 14:58:49.831 INFO 22784 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.
```