

Web API Design with Spring Boot Week 15 Coding Assignment


Points possible: 75

URL to GitHub Repository: [kmaradiaga18/Week13IntroSpringBoot](https://github.com/kmaradiaga18/Week13IntroSpringBoot): 1st week of SpringBoot (github.com)


URL to Public Link of your Video: <https://youtu.be/by34e4ZgG3Q>

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>

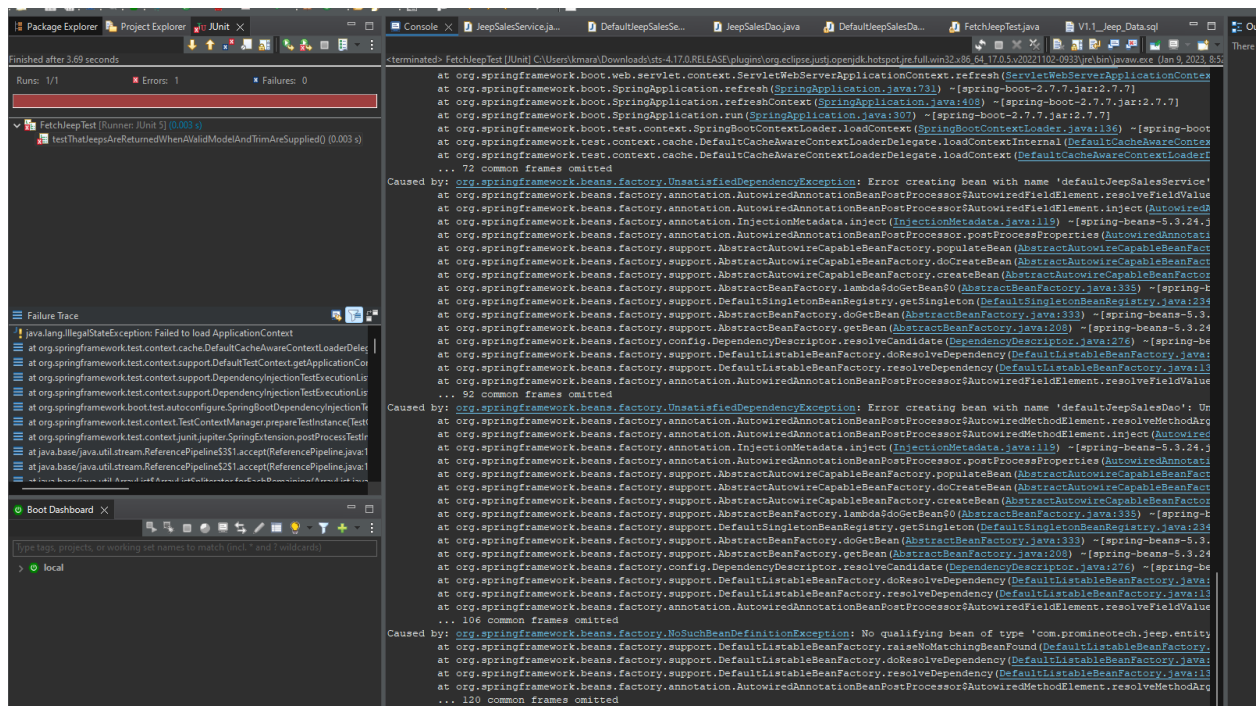
Coding Steps:

- 1) In the application you've been building add a DAO layer:
 - a) Add the package, `com.promineotech.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).

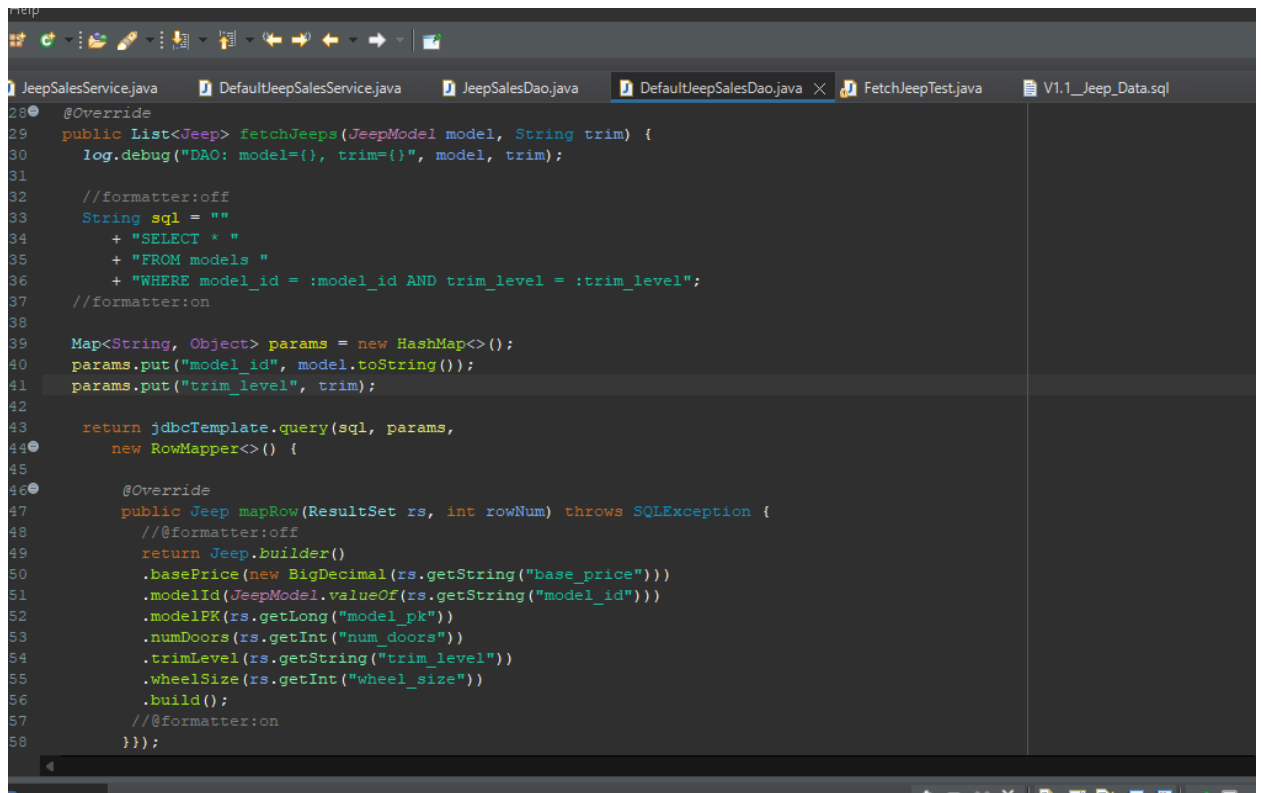
Web API Design with Spring Boot Week 15 Coding Assignment

- 3) In the DAO implementation class (DefaultJeepSalesDao):
- 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.

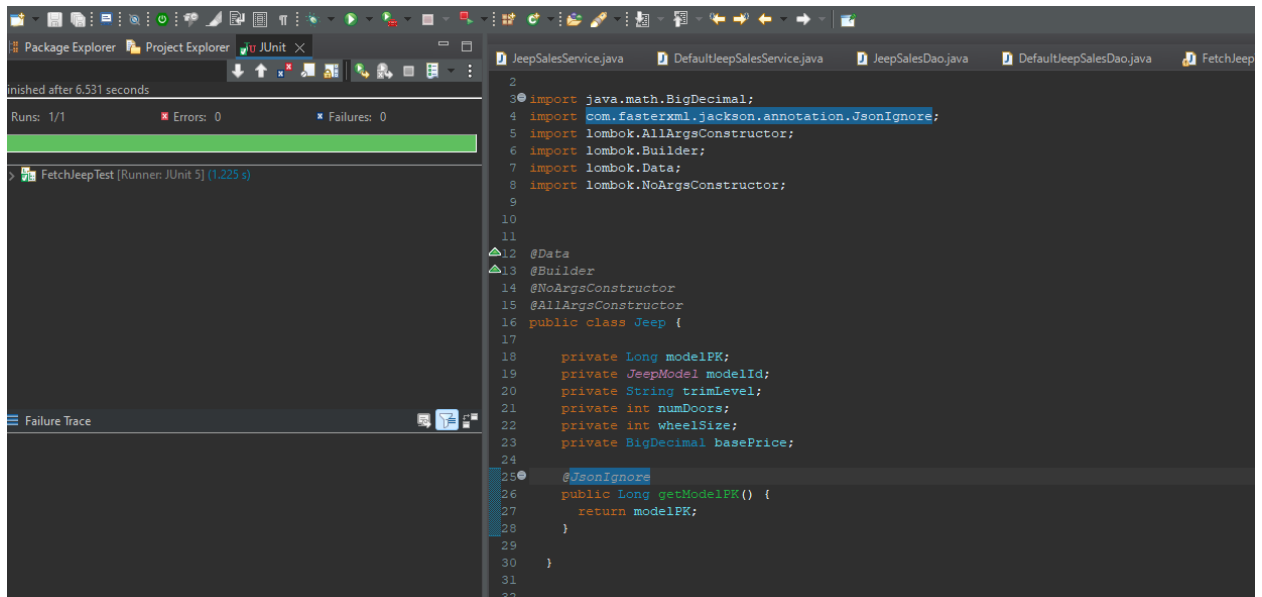
Web API Design with Spring Boot Week 15 Coding Assignment



```
28 @Override
29 public List<Jeep> fetchJeeps(JeepModel model, String trim) {
30     log.debug("DAO: model={}, trim={}", model, trim);
31
32     //formatter:off
33     String sql = "
34         + "SELECT * "
35         + "FROM models "
36         + "WHERE model_id = :model_id 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,
44         new RowMapper<>() {
45
46         @Override
47         public Jeep mapRow(ResultSet rs, int rowNum) throws SQLException {
48             //formatter:off
49             return Jeep.builder()
50                 .basePrice(new BigDecimal(rs.getString("base_price")))
51                 .modelId(JeepModel.valueOf(rs.getString("model_id")))
52                 .modelPK(rs.getLong("model_pk"))
53                 .numDoors(rs.getInt("num_doors"))
54                 .trimLevel(rs.getString("trim_level"))
55                 .wheelSize(rs.getInt("wheel_size"))
56                 .build();
57             //formatter:on
58         }});
```

- 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. 

Web API Design with Spring Boot Week 15 Coding Assignment



The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure.
- Project Explorer:** Shows the project files.
- JUnit:** A tab for running tests.
- JUnit Console:** Displays the test results for `FetchJeepTest`. It shows that the test passed after 6.531 seconds, with 1/1 runs, 0 errors, and 0 failures.
- Failure Trace:** A section for viewing test failures, currently empty.
- Source Code:** The `JeepSalesService.java` file is open, showing the `Jeep` class. The class has the following attributes and methods:

```
2
3 import java.math.BigDecimal;
4 import com.fasterxml.jackson.annotation.JsonIgnore;
5 import lombok.AllArgsConstructor;
6 import lombok.Builder;
7 import lombok.Data;
8 import lombok.NoArgsConstructor;
9
10
11
12 @Data
13 @Builder
14 @NoArgsConstructor
15 @AllArgsConstructor
16 public class Jeep {
17
18     private Long modelFK;
19     private JeepModel modelId;
20     private String trimLevel;
21     private int numDoors;
22     private int wheelSize;
23     private BigDecimal basePrice;
24
25     @JsonIgnore
26     public Long getModelFK() {
27         return modelFK;
28     }
29
30 }
31
32
```