Points possible: 75

URL to GitHub Repository: <a href="https://github.com/ssteve1/Week16--SpringBoot-JeepSales.git">https://github.com/ssteve1/Week16--SpringBoot-JeepSales.git</a>
URL to Public Link of your Video: <a href="https://youtu.be/w5e7-IDqwUI">https://youtu.be/w5e7-IDqwUI</a>

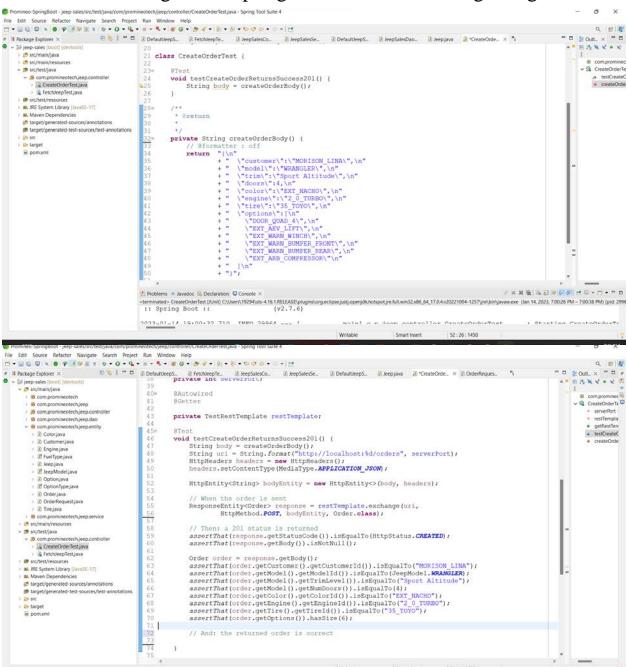
\_\_\_\_\_

#### **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.
  - <u>Easy way to Create a video</u>: 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.
  - <u>Easy way to Create a Public Video Link</u>: 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.

\_\_\_\_\_



```
File Edit Source Refactor Navigate Search Project Run Window Help

☑ JeepSalesController.java

                                                                                                             summary = "Create an order for a jeep",
description = "Returns the created Jeep",
responses = {
@ApiResponse(
                 m com.promineotech.jeep.dao
                   Control Contro
                                                                                                                                                     responseCode = "201",
description = "The created Jeep is returned",
content = @Content(mediaType = "application/json",
schema = @Schema (implementation = Order.class ))),
                    ☑ Jeepjava
☑ JeepModel.java

    DeepMounjana
    OptionType.java

                                                                                                                                 @ApiResponse(
    responseCode = "400",
    description = "The request parameters are invalid)",
    content = @Content(mediaType = "application/json")),
@ApiResponse(
    responseCode = "404",
    description = "A Jeep component was not were found in the input criteria",
    content = @Content(mediaType = "application/json")),
@ApiResponse(
                 a compromineotech jeep service
             src/main/resources
         ~ # src/test/java
                                                                                                                                 @ApiResponse(
                    CreateOrderTest.java
FetchJeepTest.java
                                                                                                                                                    responseCode = "500",
description = "An unplanned error occurred",
content = @Content(mediaType = "application/json"))
           M JRE System Library (JavaSE-17)
                                                                                                             parameters = {
           Mayen Dependencies
            a target/generated-sources/annotations
                                                                                                                               @Parameter(
            # target/generated-test-sources/test-annota
                                                                                                                                                     tor(
  name = "orderRequest",
  required = true,
  description = "The order as JSON"),
 File Edit Source Refactor Navigate Search Project Run Window Help
     9 public interface JeepOrderController (
                                                                                             //@formatter:off
                  # com.promineotech.jeep
                                                                                               @Operation(
                                                                                                                   summary = "Create an order for a jeep",
description = "Returns the created Jeep",
responses = {
@ApiResponse(

w a com.promineotech.jeep.controller

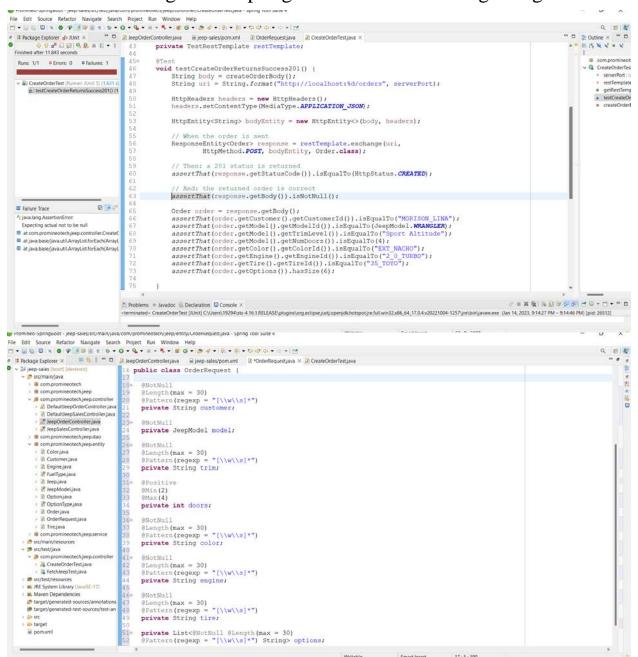
    DefaultleepSalesController.java
    BleepOrderController.java

                      III JeepSalesController.java
                                                                                                                                                             ponse(
responseCode = "201",
description = "The created Jeep is returned",
content = @Content (mediaType = "application/json",
schema = @Schema (implementation = Order.class ))),
                  m com.promineotech.ieep.dao
                   Customerjava
                      (f) Engine.iava
                                                                                                                                         @ApiResponse(

☑ FuelType.java
☑ Jeep.java

                                                                                                                                                              ponse(
responseCode = "400",
description = "The request parameters are invalid)",
content = @Content(mediaType = "application/json")),
                      III JeepModel.java
                   Doption.java
                                                                                                                                         @ApiResponse(
                                                                                                                                                              ponse(
responseCode = "404",
description = "A Jeep component was not were found in the input criteria",
content = @Content(mediaType = "application/json")),
                    Order.java
                      OrderRequest.java
                      (i) Tire.iava
                                                                                                                                         @ApiResponse(
                                                                                                                                                              ponse(
responseCode = "500",
description = "An unplanned error occurred",
content = @Content(mediaType = "application/json"))
          CreateOrderTest.java
                                                                                                                    parameters = {
             src/test/resources
             M JRE System Library [JavaSE-17]
                                                                                                                                                              name = "orderRequest",
             Mayen Dependencies
target/generated-sources/annotations
                                                                                                                                                              required = true,
description = "The order as JSON"),
             # target/generated-test-sources/test-annota
                                                                                                //@formatter: on
                                                                                               4PostMapping

4ResponseStatus(code = HttpStatus.CREATED)
             pom.xml
                                                                                                Order createOrder(@RequestBody OrderRequest orderRequest );
```



```
E % | □ □ | Defaultle
                                                                                         package com.promineotech.jeep.dao;
          DefaultleepSalesController.java
                                                                                       3*import java.math.BigDecimal;

☑ JeepOrderController.iava

          JeepSalesController.iava
          DefaultleepOrderDao.java
DefaultleepOrderDao.java
DefaultleepSalesDao.java
DefaultleepSalesDao.java
DepSalesDao.java
                                                                                         public class DefaultJeepOrderDao implements JeepOrderDao {
                                                                                                   private NamedParameterJdbcTemplate jdbcTemplate;
        Color.java
                                                                                                    41
42
43
44
45
46
47
48
49
50
          @ Option java
                                                                                                      KeyHolder keyHolder = new GeneratedKeyHolder();
          P OptionType.java
                                                                                                      jdbcTemplate.update(params.sql, params.source,keyHolder);
                                                                                                      Long orderPk = keyHolder.getKey().longValue();
saveOptions(options, orderPk);

    D Default/LepSalesSenrice Java
    D I Default/LepSalesSenrice Java
    D I Deathboard ×
                                                                                                      return Order.builder()
                                                                                                                  Order. Dullder()
.orderPK(orderPk)
.customer(custome
.model(jeep)
.color(color)
.engine(engine)
.tire(tire)
.options(options)
 Type tags, projects, or working set names to match (incl. * and ? wildcards)
                                                                                                                    .price(price)
.build();
I elements hidden by filter
File Edit Source Refactor Navigate Search Project Run Window Help
                                   E % 1 <sup>™</sup> □ ② ReepOrderSe. ② Pefaultieep. × ② Defaultieep. □ Defaultieep. ② Defaultieep. ② Defaultieep. ② DepSalesDao. ② DeepSalesDao. ② ReepOrderDao. ③ ReepOrderDao.
  private JeepOrderDao jeepOrderDao;
      # com.promineotech
       # com.promineotech.jeep

compromineotech jeep

Ecompromineotech jeep controller

Default Jeep Crder Controller java

Default Jeep Sales Controller java

Papp Order Controller java

If Jeep Sales Controller java
                                                                                                public Order createOrder(OrderRequest orderRequest) (
                                                                                                      log.debug("Order = ()", orderRequest);
                                                                                                      Customer customer = getCustomer(orderRequest);

Jeep jeep = getModel(orderRequest);

Color color = getColor(orderRequest);

Engine engine = getEngine(orderRequest);

Tire tire = getTire(orderRequest);
     ~ # com.promineotech.ieep.dac
        > In DefaultJeepOrderDao.java

DefaultleepDalesDaojava
DefaultleepDalesDaojava
DepOrderDaojava
RepOrderDaojava
Compromineotechjeep.entity
                                                                                                      BigDecimal price = jeep.getBasePrice().add(color.getPrice())
                                                                                                                    .add(engine.getPrice()).add(tire.getPrice());

DefaultbeepOrderService.java
DefaultbeepSalesService.java
DepSalesService.java
Strc/main/resources
Strc/test/java
Depositionatoric java
Depositionatoric java
Depositionatoric java
Depositionatoric java
Depositionatoric java
                                                                                                      return jeepOrderDao.saveOrder(customer, jeep, color, engine, tire, price, null);
                                                                                                    * @param orderRequest
* @return
       → A CreateOrderTest.java
private List<Option> getOption(OrderRequest orderRequest) {
   return jeepOrderDao.fetchOptions(orderRequest.getOptions());
  @ local
                                                                                                  elements hidden by filter
```

