

URL to GitHub Repository: <https://github.com/aliustunyer/Web-API-Design-with-Spring-Boot.git>

URL to Youtube Video : <https://youtu.be/1BWjq4WYwqw>

```
package com.promineotech.jeepp;
```

```
import org.springframework.boot.SpringApplication;  
import  
org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication  
public class jeepSales {
```

```
    public static void main(String[] args) {  
        SpringApplication.run(jeepp.class, args);  
    }  
}
```

```
package com.promineotech.jeepp.controller;
```

```
package com.promineotech.jeepp.controller;
```

```
import java.util.List;  
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.web.bind.annotation.RestController;
import com.promineotech.jee.entity.Jee;
import com.promineotech.jee.service.JeeSalesService;
import lombok.extern.slf4j.Slf4j;
```

```
@RestController
```

```
@Slf4j
```

```
public class DefaultJeeSalesController implements
JeeSalesController {
```

```
@Autowired
```

```
private JeeSalesService jeeSalesService;
```

```
@Override
```

```
public List<Jee> fetchJeeps(String model, String trim) {
log.info("model = {}, trim = {}", model, trim);
return jeeSalesService.fetchJeeps(model, trim);
}
```

```
}
```

```
package com.promineotech.jee.controller;
```

```
import java.util.List;
```

```
import org.springframework.http.HttpStatus;
```

```
import org.springframework.web.bind.annotation.GetMapping;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseStatus;
import com.promineotech.jee.entity.Jee;
import io.swagger.v3.oas.annotations.OpenAPIDefinition;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.Parameter;
import io.swagger.v3.oas.annotations.info.Info;
import io.swagger.v3.oas.annotations.media.Content;
import io.swagger.v3.oas.annotations.media.Schema;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.servers.Server;
```

```
@RequestMapping("/jeeps")
```

```
@OpenAPIDefinition(info = @Info(title = "Jee Sales Service"),
servers = {
```

```
@Server(url = "http://localhost:8080", description = "Local server."))
```

```
public interface JeeSalesController {
```

```
//@formatter:off
```

```
@Operation(
```

```
summary = "Returns a list of Jeeps",
```

```
description = "Returns a list of Jeeps given an optional model and/or trim",
```

```
responses = {
```

```
@ApiResponse(
```

```
responseCode="200",
description = "A list of Jeeps is returned",
content = @Content(
mediaType = "application/json",
schema = @Schema(implementation = Jeep.class))),
@ApiResponse(
responseCode = "400",
description = "The request parameters are invalid",
content = @Content(mediaType = "application/json")),
@ApiResponse(
responseCode = "404",
description = "No Jeeps were found with the input criteria",
content = @Content(mediaType = "application/json")),
@ApiResponse(
responseCode = "500",
description = "An unplanned error occurred",
content = @Content(mediaType = "application/json")),
},
parameters = {
@Parameter(name = "model",
allowEmptyValue = false,
required = false,
description = "The model name (i.e., 'WRANGLER')"),
@Parameter(name = "trim",
allowEmptyValue = false,
```

```

        required = false,
        description = "The trim level (i.e., 'Sport')"),
    }
)
@GetMapping
@ResponseStatus(code = HttpStatus.OK)
List<Jeep> fetchJeeps(
    @RequestParam(required = false)
    String model,
    @RequestParam(required = false)
    String trim);
//@formatter: on
}

```

```

package com.promineotech.jeepproject.service;

import java.util.List;
import org.springframework.stereotype.Service;
import com.promineotech.jeepproject.entity.Jeep;
import lombok.extern.slf4j.Slf4j;

@Service
@Slf4j
public class DefaultJeepSalesService implements JeepSalesService {

    @Override
    public List<Jeep> fetchJeeps(String model, String trim) {
        log.info("The fetchJeeps method was called with model={} and trim={}",
            model, trim);
        return null;
    }
}

```

```
package com.promineotech.jeepproject.service;

import java.util.List;
import com.promineotech.jeepproject.entity.Jeepproject;

public interface JeepSalesService {

    List<Jepproject> fetchJeeps(String model, String trim);

}
```

application.yaml

```
spring:
  datasource:
    username: .....
    password: .....
    url: jdbc:mysql://localhost:3306/jeepproject
```

```
package com.promineotech.jeepproject.controller;

import static org.assertj.core.api.Assertions.assertThat;
import static org.junit.jupiter.api.Assertions.*;
import java.util.List;
import java.math.BigDecimal;
import java.util.LinkedList;

import org.junit.jupiter.api.Test;
```

```

import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;
import org.springframework.core.ParameterizedTypeReference;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import com.promineotech.jeeep.controller.support.FetchJeepTestSupport;
import com.promineotech.jeeep.entity.JeeepModel;
import com.promineotech.jeeep.entity.Jeeep;

```

```

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
class FetchJeepTest extends FetchJeepTestSupport{

@Test
void testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied() {
//Given : a valid model, trim and URI
JeeepModel model = JeeepModel.WRANGLER;
String trim = "Sport";
String uri =
String.format ("%s?model=%s&trim=%s", getBaseUri(), model, trim);
//System.out.println(uri);
//When : a connection is made to the URI
ResponseEntity<List<Jeeep>> response =
getRestTemplate().exchange(uri, HttpMethod.GET, null, new
ParameterizedTypeReference<>() {});
//Then : a success (OK-200) status code is returned
assertThat(response.getStatusCode()).isEqualTo(HttpStatus.OK);
//And : the actual list returned is the same as the expected list
List<Jeeep> expected = buildExpected();
assertThat(response.getBody()).isEqualTo(expected);
}

```

```

private List<Jeeep> buildExpected() {
List<Jeeep> list = new LinkedList<>();
//@formatter : off
list.add(Jeeep.builder()

```

```

        .modelId(JeepModel.WRANGLER)
        .trimLevel("Sport")
        .numDoors(2)
        .wheelSize(17)
        .basePrice(new BigDecimal("28475.00"))
        .build());
list.add(Jeep.builder()
        .modelId(JeepModel.WRANGLER)
        .trimLevel("Sport")
        .numDoors(4)
        .wheelSize(17)
        .basePrice(new BigDecimal("31975.00"))
        .build());
//@formatter : off
return list;
}

}

```

POM

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0
    .0 https://maven.apache.org/xsd/maven-4.0.0.xsd>
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.6.14</version>

```



```
<relativePath/> <!-- lookup parent from repository
-->
```

```
</parent>
```

```
<groupId>com.promineotech</groupId>
```

```
<artifactId>jeep-sales</artifactId>
```

```
<version>1.0.0.1-SNAPSHOT</version>
```

```
<name>jeep-sales</name>
```

```
<description>Jeep Sales</description>
```

```
<properties>
```

```
<java.version>17</java.version>
```

```
</properties>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-web</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-devtools</artifactId>
```

```
<scope>runtime</scope>
```

```
<optional>true</optional>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.projectlombok</groupId>
```

```
<artifactId>lombok</artifactId>
```

```
<optional>true</optional>
</dependency>
<dependency>
<groupId>org.springdoc</groupId>
<artifactId>springdoc-openapi-ui</artifactId>
<version>1.5.5</version>
</dependency>
<!-- Database dependencies
===== -->

<!--
https://mvnrepository.com/artifact/org.springframework
ork.boot/spring-boot-starter-jdbc -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>

<!--
https://mvnrepository.com/artifact/com.mysql/mysql-
connector-j -->
<dependency>
<groupId>com.mysql</groupId>
<artifactId>mysql-connector-j</artifactId>
</dependency>

<!-- Test dependencies
===== -->
```

```
<!--  
https://mvnrepository.com/artifact/com.h2database/h2 -->
```

```
<!--  
https://mvnrepository.com/artifact/com.h2database/h2 -->
```

```
<dependency>  
  <groupId>com.h2database</groupId>  
  <artifactId>h2</artifactId>  
  <scope>test</scope>  
</dependency>
```

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-test</artifactId>  
  <scope>test</scope>  
</dependency>  
</dependencies>
```

```
<build>  
  <plugins>  
    <plugin>  
      <groupId>org.springframework.boot</groupId>  
      <artifactId>spring-boot-maven-plugin</artifactId>  
      <configuration>  
        <excludes>
```

```
<exclude>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
</exclude>
</excludes>
</configuration>
</plugin>
</plugins>
</build>

</project>
```