

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

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

```
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;
```

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

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

```
import com.promineotech.jeepp.entity.Jeepp;
```

```
@RestController
```

```
public class DefaultJeepSalesController implements JeepSalesController {
```

```
    @Override
```

```
    public List<Jeep> fetchJeeps(String model, String trim) {
```

```
        return null;
```

```
    }
```

```
}
```

```
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.jeepp.entity.Jeepp;
```

```
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 = "Jeep Sales Service"), servers = {
    @Server(url = "http://localhost:8080", description = "Local server.")})
```

```
public interface JeepSalesController {

    // @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.jeep.controller;

import static org.assertj.core.api.Assertions.assertThat;
import java.util.List;
import org.junit.jupiter.api.Test;
import org.springframework.beans.factory.annotation.Autowired;

```

```
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.ActiveProfiles;
import org.springframework.boot.test.context.SpringBootTest.WebEnvironment;
import org.springframework.boot.test.web.client.TestRestTemplate;
import org.springframework.boot.web.server.LocalServerPort;
import org.springframework.core.ParameterizedTypeReference;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.test.context.jdbc.Sql;
import org.springframework.test.context.jdbc.SqlConfig;
import com.promineotech.jee.entity.Jee;
import com.promineotech.jee.entity.JeeModel;

import org.springframework.http.HttpMethod;
```

```
@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
```

```
@ActiveProfiles("test")
```

```
@Sql(scripts = {
    "classpath:flyway/migrations/V1.0__Jee_Schema.sql",
    "classpath:flyway/migrations/V1.1__Jee_Data.sql"},
```

```
config = @SqlConfig(encoding = "utf-8"))
```

```
class FetchJeeTest2 {

    @Autowired
    private TestRestTemplate restTemplate;

    @LocalServerPort
    private int serverPort;
```

```

@Test
void testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied() {
    JeepModel model = JeepModel.WRANGLER;
    String trim = "Sport";
    String uri =
        String.format("http://localhost:%d/jeeps?model=%s&trim=%s", serverPort, model, trim);

    ResponseEntity<List<Jeep>> response =
        restTemplate.exchange(uri, HttpMethod.GET, null, new ParameterizedTypeReference<>() {});

    assertThat(response.getStatusCode()).isEqualTo(HttpStatus.OK);
}
}

```

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>jeeo-sales</artifactId>
<version>1.0.0.1-SNAPSHOT</version>
<name>jeeo-sales</name>
<description>Jeeo 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>
<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>
```