

<https://youtu.be/acC6xf4rkJY>
[git@github.com:lixyl979/java_week_13.git](https://github.com:lixyl979/java_week_13.git)

```
package com.promineotech.jee;

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

@SpringBootApplication(scanBasePackages = {"com.promineotech"})
public class JeepSales {

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

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

import static org.assertj.core.api.Assertions.assertThat;

import java.math.BigDecimal;
import java.util.LinkedList;
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.boot.test.context.SpringBootTest.WebEnvironment;
import org.springframework.boot.test.web.client.TestRestTemplate;
import org.springframework.boot.test.web.server.LocalServerPort;
import org.springframework.core.ParameterizedTypeReference;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.test.context.ActiveProfiles;
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 lombok.Getter;

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@ActiveProfiles("test")
@Sql(scripts = {
    "classpath:flyway/migrations/V1.0__Jee_Schema.sql",
```

```

        "classpath:flyway/migrations/V1.1__Jeep_Data.sql"},
        config = @SqlConfig(encoding = "utf-8"))
class FetchJeepTest {
    @LocalServerPort
    private int serverPort;

    @Autowired
    @Getter
    private TestRestTemplate restTemplate;

    protected List<Jeep> buildExpected(){
        List<Jeep> list = new LinkedList<>();
        //@formatter:off
        list.add(Jeep.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:on
        return list;
    }

    @Test
    void testThatJeepsAreReturnedWhenAValidModelAndTrimAreSupplied() {
        // Given: a valid model, trim and URI
        JeepModel model = JeepModel.WRANGLER;
        String trim = "Sport";
        String uri =
            String.format("http://localhost:%d/jeeps?model=%s&trim=%s", serverPort,
model, trim);
        System.out.println(uri);
        // When: a connection is made to the URI

        ResponseEntity<List<Jeep>> response =
            restTemplate.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<Jeep> expected = buildExpected();
        System.out.println(expected);
        assertThat(response.getBody()).isEqualTo(expected);
    }
}

package com.promineotech.jeep.service;

```

```

import java.util.List;
import org.springframework.stereotype.Service;
import com.promineotech.jeeep.entity.Jeeep;
import com.promineotech.jeeep.entity.JeeepModel;
import lombok.extern.slf4j.Slf4j;

@Service
@Slf4j
public class DefaultJeeepSalesService implements JeeepSalesService {

    @Override
    public List<Jeeep> fetchJeeeps(JeeepModel model, String trim) {
        log.info("The fetchJeeeps method was called with model = {} "
            + "and trim = {}", model, trim);

        return null;
    }
}

```

```

package com.promineotech.jeeep.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.RestController;

import com.promineotech.jeeep.entity.Jeeep;
import com.promineotech.jeeep.entity.JeeepModel;
import com.promineotech.jeeep.service.JeeepSalesService;

import lombok.extern.slf4j.Slf4j;

@RestController
@Slf4j
public class DefaultJeeepSalesController implements JeeepSalesController {

    @Autowired
    private JeeepSalesService jeeepSalesService;

    @Override
    public List<Jeeep> fetchJeeeps(JeeepModel model, String trim) {
        log.debug("model = {}, trim = {}", model, trim);
        return jeeepSalesService.fetchJeeeps(model, trim);
    }
}

```

```

spring:
datasource:
password: jeeep
username: jeeep
url: jdbc:mysql://localhost:3306/jeeep

```

```

spring:
datasource:

```

```
url: jdbc:h2:mem:jeep;mode=MySQL
```

```
<?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>2.7.6</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>
<!-- Bean Validation ===== -->
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-validation -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-validation</artifactId>
</dependency>

<!-- OpenAPI dependency ===== -->
<!-- https://mvnrepository.com/artifact/org.springdoc/springdoc-openapi-ui -->
<dependency>
<groupId>org.springdoc</groupId>
<artifactId>springdoc-openapi-ui</artifactId>
<version>1.6.13</version>
</dependency>

<!-- Database Ddependencies ===== -->
<!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-data-jdbc -->
<dependency>
```

```

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-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 -->
<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>

```

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

```

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",
                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,
description = "The model name

```
(i.e., 'Sport'))  
        }  
    )  
    @GetMapping  
    @ResponseStatus(code = HttpStatus.OK)  
    List<Jeep> fetchJeeps(  
        @RequestParam JeepModel model,  
        @RequestParam String trim);  
    // @formatter:on  
}
```

```
package com.promineotech.jeep.entity;
```

```
import java.math.BigDecimal;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Builder;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
@Data
```

```
@Builder
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
public class Jeep {  
    private Long modelPK;  
    private JeepModel modelId;  
    private String trimLevel;  
    private int numDoors;  
    private int wheelSize;  
    private BigDecimal basePrice;  
}
```

```
package com.promineotech.jeep.entity;
```

```
public enum JeepModel {  
    WRANGLER, GRAND_CHEROKEE, CHEROKEE, COMPASS, RENEGADE, GLADIATOR,  
    WRANGLER_4XE  
}
```

```
package com.promineotech.jeep.service;
```

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

```
import com.promineotech.jeep.entity.Jeep;
```

```
import com.promineotech.jeep.entity.JeepModel;
```

```
public interface JeepSalesService {
```

```
    /*  
    * @param model  
    * @param trim  
    * @return  
    */
```

```
List<Jeep> fetchJeeps(JeepModel model, String trim);
```

```
}
```