

[https://github.com/lixy1979/java\\_week\\_13](https://github.com/lixy1979/java_week_13)  
<https://youtu.be/Q0DpJksWE0w>

```
package com.promineotech.jeepp;

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.jeepp.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.jeepp.entity.Jeep;
import com.promineotech.jeepp.entity.JeepModel;

import lombok.Getter;

@SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
@ActiveProfiles("test")
@Sql(scripts = {
    "classpath:flyway/migrations/V1.0__Jeep_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();

    assertThat(response.getBody()).isEqualTo(expected);
}
}

package com.promineotech.jeep.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.promineotech.jeep.dao.JeepSalesDao;

```

```

import com.promineotech.jeepp.entity.Jeepp;
import com.promineotech.jeepp.entity.JeeppModel;
import lombok.extern.slf4j.Slf4j;

@Service
@Slf4j
public class DefaultJeepSalesService implements JeepSalesService {

    @Autowired
    private JeepSalesDao jeepSalesDao;

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

package com.promineotech.jeepp.service;

import java.util.List;

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

public interface JeepSalesService {

    /**
     * @param model
     * @param trim
     * @return
     */
    List<Jeep> fetchJeeps(JeeppModel model, String trim);
}

package com.promineotech.jeepp.dao;

import java.util.List;

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

public interface JeepSalesDao {

    /**
     *
     * @param model
     * @param trim
     * @return
     */
    List<Jeep> fetchJeeps(JeeppModel model, String trim);
}

```

```

}

package com.promineotech.jeepp.dao;

import java.math.BigDecimal;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.RowMapper;
import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
import org.springframework.stereotype.Component;

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

import lombok.extern.slf4j.Slf4j;

@Component
@Slf4j
public class DefaultJeeppSalesDao implements JeeppSalesDao {

    @Autowired
    private NamedParameterJdbcTemplate jdbcTemplate;

    @Override
    public List<Jeepp> fetchJeepps(JeeppModel model, String trim){
        log.debug("DAO: model={}, trim={}", model, trim);
        // @formatter: off
        String sql = ""
            + "SELECT * "
            + "FROM models "
            + "WHERE model_id = :model_id AND trim_level = :trim_level";
        // @formatter: on
        Map<String, Object> params = new HashMap<>();
        params.put("model_id", model.toString());
        params.put("trim_level", trim);
        return jdbcTemplate.query(sql, params, new RowMapper<>(){

            @Override
            public Jeepp mapRow(ResultSet rs, int rowNum) throws SQLException {
                //@formatter:off
                return Jeepp.builder()
                    .basePrice(new BigDecimal(rs.getString("base_price")))
                    .modelId(JeeppModel.valueOf(rs.getString("model_id")))
                    .modelPK(rs.getLong("model_Pk"))
                    .numDoors(rs.getInt("num_doors"))
                    .trimLevel(rs.getString("trim_level"))
                    .wheelSize(rs.getInt("wheel_size"))
                    .build();
                //@formatter:on
            }
        });
    }
}

```

```

}

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.jeepp.entity.Jeepp;
import com.promineotech.jeepp.entity.JeeppModel;
import com.promineotech.jeepp.service.JeeppSalesService;

import lombok.extern.slf4j.Slf4j;

@RestController
@Slf4j
public class DefaultJeeppSalesController implements JeeppSalesController {

    @Autowired
    private JeeppSalesService jeeppSalesService;

    @Override
    public List<Jeepp> fetchJeepps(JeeppModel model, String trim) {
        log.debug("model = {}, trim = {}", model, trim);
        return jeeppSalesService.fetchJeepps(model, trim);
    }
}

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("/jeepps")
    @OpenAPIDefinition(info = @Info(title = "Jeepp sales service"), servers = {
        @Server(url = "http://localhost:8080",description = "local server.")})
    public interface JeeppSalesController {

```

```

//@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 java.util.Comparator;

import com.fasterxml.jackson.annotation.JsonIgnore;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class Jeep implements Comparable<Jeep>{
    private Long modelPK;
    private JeepModel modelId;
    private String trimLevel;
    private int numDoors;
    private int wheelSize;
    private BigDecimal basePrice;

    @JsonIgnore
    public Long getModelPK() {
        return modelPK;
    }

    @Override
    public int compareTo(Jeep that) {
        //@formatter:off
        return Comparator
            .comparing(Jeep::getModelId)
            .thenComparing(Jeep::getTrimLevel)
            .thenComparing(Jeep::getNumDoors)
            .compare(this, that);
        //@formatter:on
    }
}

package com.promineotech.jeep.entity;

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

```