

https://github.com/lixy1979/java_week_13

<https://youtu.be/6yxHOKxVi6Q>

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

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

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.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
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.JeeModel;
import com.promineotech.jee.entity.Order;

@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"))
public class CreateOrderTest {

    @LocalServerPort
    private int serverPort;

    @Autowired
    private TestRestTemplate restTemplate;

    @Test
    void testCreateOrderReturnsSuccess201() {
        //Given: an order as JSON
        String body = createOrderBody();
        String uri = String.format("http://localhost:%d/orders", serverPort);

        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.APPLICATION_JSON);

        HttpEntity<String> bodyEntity = new HttpEntity<>(body, headers);

        //When: the order is sent
        ResponseEntity<Order> response = restTemplate.exchange(uri,
            HttpMethod.POST, bodyEntity, Order.class);
    }
}
```

```

//Then: a 201 status is returned
assertThat(response.getStatusCode()).isEqualTo(HttpStatus.CREATED);

//And: the returned order is correct
assertThat(response.getBody()).isNotNull();

Order order = response.getBody();
assertThat(order.getCustomer().getCustomerId()).isEqualTo("MORISON_LINA");
assertThat(order.getModel().getModelId()).isEqualTo(JeepModel.WRANGLER);
assertThat(order.getModel().getTrimLevel()).isEqualTo("Sport Altitude");
assertThat(order.getModel().getNumDoors()).isEqualTo(4);
assertThat(order.getColor().getColorId()).isEqualTo("EXT_NACHO");
assertThat(order.getEngine().getEngineId()).isEqualTo("2_0_TURBO");
assertThat(order.getTire().getTireId()).isEqualTo("35_TOYO");
assertThat(order.getOptions()).hasSize(6);
}

protected String createOrderBody() {
// @formatter:off
    return "{\n"
        + "  \"customer\": \"MORISON_LINA\", \n"
        + "  \"model\": \"WRANGLER\", \n"
        + "  \"trim\": \"Sport Altitude\", \n"
        + "  \"doors\": 4, \n"
        + "  \"color\": \"EXT_NACHO\", \n"
        + "  \"engine\": \"2_0_TURBO\", \n"
        + "  \"tire\": \"35_TOYO\", \n"
        + "  \"options\": [\n"
        + "    \"DOOR_QUAD_4\", \n"
        + "    \"EXT_AEV_LIFT\", \n"
        + "    \"EXT_WARN_WINCH\", \n"
        + "    \"EXT_WARN BUMPER_FRONT\", \n"
        + "    \"EXT_WARN BUMPER_REAR\", \n"
        + "    \"EXT_ARB_COMPRESSOR\" \n"
        + "  ] \n"
        + "}";
// @formatter:on
}
}

```

```
package com.promineotech.jeep.dao;
```

```

import java.math.BigDecimal;
import java.util.List;
import java.util.Optional;
import com.promineotech.jeep.entity.Color;
import com.promineotech.jeep.entity.Customer;
import com.promineotech.jeep.entity.Engine;
import com.promineotech.jeep.entity.Jeep;
import com.promineotech.jeep.entity.JeepModel;
import com.promineotech.jeep.entity.Option;
import com.promineotech.jeep.entity.Order;
import com.promineotech.jeep.entity.Tire;

```

```
public interface JeepOrderDao {
```

```

List<Option> fetchOptions(List<String> optionIds);
Optional<Customer> fetchCustomer(String customerId);
Optional<Jeep> fetchModel(JeepModel model, String trim, int doors);
Optional<Color> fetchColor(String colorId);
Optional<Engine> fetchEngine(String engineId);
Optional<Tire> fetchTire(String tireId);
Order saveOrder(Customer customer, Jeep jeep, Color color, Engine engine,
                Tire tire, BigDecimal Price, List<Option> options);
}

```

```

package com.promineotech.jeep.dao;

```

```

import java.math.BigDecimal;
import java.util.List;
import java.util.Optional;
import com.promineotech.jeep.entity.Color;
import com.promineotech.jeep.entity.Customer;
import com.promineotech.jeep.entity.Engine;
import com.promineotech.jeep.entity.Jeep;
import com.promineotech.jeep.entity.JeepModel;
import com.promineotech.jeep.entity.Option;
import com.promineotech.jeep.entity.Order;
import com.promineotech.jeep.entity.Tire;

public interface JeepOrderDao {
    List<Option> fetchOptions(List<String> optionIds);
    Optional<Customer> fetchCustomer(String customerId);
    Optional<Jeep> fetchModel(JeepModel model, String trim, int doors);
    Optional<Color> fetchColor(String colorId);
    Optional<Engine> fetchEngine(String engineId);
    Optional<Tire> fetchTire(String tireId);
    Order saveOrder(Customer customer, Jeep jeep, Color color, Engine engine,
                    Tire tire, BigDecimal Price, List<Option> options);
}

```

```

package com.promineotech.jeep.service;

```

```

import com.promineotech.jeep.entity.Order;
import com.promineotech.jeep.entity.OrderRequest;

```

```

public interface JeepOrderService {

    Order createOrder(OrderRequest orderRequest);

}

```

```

package com.promineotech.jeep.service;

```

```

import java.math.BigDecimal;
import java.util.List;
import java.util.NoSuchElementException;

```

```

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

import com.promineotech.jeepp.dao.JeeppOrderDao;
import com.promineotech.jeepp.entity.Color;
import com.promineotech.jeepp.entity.Customer;
import com.promineotech.jeepp.entity.Engine;
import com.promineotech.jeepp.entity.Jeepp;
import com.promineotech.jeepp.entity.Option;
import com.promineotech.jeepp.entity.Order;
import com.promineotech.jeepp.entity.OrderRequest;
import com.promineotech.jeepp.entity.Tire;

@Service
public class DefaultJeeppOrderService implements JeeppOrderService {

    @Autowired
    private JeeppOrderDao jeeppOrderDao;

    @Transactional
    @Override
    public Order createOrder(OrderRequest orderRequest) {
        Customer customer = getCustomer(orderRequest);
        Jeepp jeep = getModel(orderRequest);
        Color color = getColor(orderRequest);
        Engine engine = getEngine(orderRequest);
        Tire tire = getTire(orderRequest);
        List<Option> options = getOption(orderRequest);
        BigDecimal price = jeep.getBasePrice().add(color.getPrice())
            .add(engine.getPrice()).add(tire.getPrice());
        for(Option option : options) {
            price = price.add(option.getPrice());
        }
        return jeeppOrderDao.saveOrder(customer, jeep, color, engine, tire, price, options);
    }

    /**
     *
     * @param orderRequest
     * @return
     */
    private List<Option> getOption(OrderRequest orderRequest) {
        return jeeppOrderDao.fetchOptions(orderRequest.getOptions());
    }

    /**
     *
     * @param orderRequest
     * @return
     */
    private Tire getTire(OrderRequest orderRequest) {
        return jeeppOrderDao.fetchTire(orderRequest.getTire())
            .orElseThrow(() -> new NoSuchElementException(
                "Tire with ID=" + orderRequest.getTire() + " was not found"));
    }
}

```

```

/**
 *
 * @param orderRequest
 * @return
 */
private Engine getEngine(OrderRequest orderRequest) {
    return jeepOrderDao.fetchEngine(orderRequest.getEngine())
        .orElseThrow(() -> new NoSuchElementException(
            "Engine with ID=" + orderRequest.getEngine() + " was not found"));
}

/**
 *
 * @param orderRequest
 * @return
 */
private Color getColor(OrderRequest orderRequest) {
    return jeepOrderDao.fetchColor(orderRequest.getColor())
        .orElseThrow(() -> new NoSuchElementException(
            "Color with ID=" + orderRequest.getColor() + " was not found"));
}

/**
 *
 * @param orderRequest
 * @return
 */
private Jeep getModel(OrderRequest orderRequest) {
    return jeepOrderDao
        .fetchModel(orderRequest.getModel(), orderRequest.getTrim(),
            orderRequest.getDoors())
        .orElseThrow(() -> new NoSuchElementException("Model with ID="
            + orderRequest.getModel() + ", trim=" + orderRequest.getTrim()
            + orderRequest.getDoors() + " was not found"));
}

/**
 *
 * @param orderRequest
 * @return
 */
private Customer getCustomer(OrderRequest orderRequest) {
    return jeepOrderDao.fetchCustomer(orderRequest.getCustomer())
        .orElseThrow(() -> new NoSuchElementException("Customer with ID="
            + orderRequest.getCustomer() + " was not found"));
}
}

```

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

```
import javax.validation.Valid;
```

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

```
import org.springframework.validation.annotation.Validated;
```

```

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;

import com.promineotech.jeepp.entity.Order;
import com.promineotech.jeepp.entity.OrderRequest;

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("/orders")
    @OpenAPIDefinition(info = @Info(title = "Jeep order service"), servers = {
        @Server(url = "http://localhost:8080",description = "local server.")})
    @Validated
public interface JeepOrderController {
    // @formatter:off
    @Operation(
        summary = "Create an order for a jeep",
        description = "returns the created jeep",
        responses = {
            @ApiResponse(
                responseCode = "201",
                description = "The created jeep is
returned",
                content = @Content(mediaType =
"application/json",
                schema = @Schema(implementation =
Order.class))),
            @ApiResponse(
                responseCode = "400",
                description = "The request parameters
are invalid",
                content = @Content(mediaType =
"application/json")
            ),
            @ApiResponse(
                responseCode = "404",
                description = "A jeep component was
not 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 = "orderRequest",
                required = true,
                description = "The order as JSON"),
        }
    )
    @PostMapping
    @ResponseStatus(code = HttpStatus.CREATED)
    Order createOrder(
        @Valid@RequestBody OrderRequest orderRequest
    );
    // @formatter:on
}

```

```

package com.promineotech.jee.controller;

```

```

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

```

```

import com.promineotech.jee.entity.Order;
import com.promineotech.jee.entity.OrderRequest;
import com.promineotech.jee.service.JeeOrderService;

```

```

import lombok.extern.slf4j.Slf4j;

```

```

@RestController
@Slf4j
public class DefaultJeeOrderController implements JeeOrderController {

    @Autowired
    private JeeOrderService jeeOrderService;

    @Override
    public Order createOrder(OrderRequest orderRequest) {
        log.debug("Order={}", orderRequest);
        return jeeOrderService.createOrder(orderRequest);
    }
}

```

```

package com.promineotech.jee.entity;

```

```

import java.util.List;

```

```

import javax.validation.constraints.Max;
import javax.validation.constraints.Min;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
import javax.validation.constraints.Positive;

```

```

import org.hibernate.validator.constraints.Length;

```

```

import lombok.Data;

```

```

@Data
public class OrderRequest {

    @NotNull
    @Length(max = 30)
    @Pattern(regexp = "[\\w\\s]*")
    private String customer;

    @NotNull
    private JeepModel model;

    @NotNull
    @Length(max = 30)
    @Pattern(regexp = "[\\w\\s]*")
    private String trim;

    @Positive
    @Min(2)
    @Max(4)
    private int doors;

    @NotNull
    @Length(max = 30)
    @Pattern(regexp = "[\\w\\s]*")
    private String color;

    @NotNull
    @Length(max = 30)
    @Pattern(regexp = "[\\w\\s]*")
    private String engine;

    @NotNull
    @Length(max = 30)
    @Pattern(regexp = "[\\w\\s]*")
    private String tire;

    private List<@NotNull @Length(max = 30) @Pattern(regexp = "[\\w\\s]*")String> options;
}

```