УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Рефакторинг баз данных и приложений»

Этап №1

Рефакторинг существующего проекта

Студенты

Белогаев Д. В.

Кузнецов М. А.

Р34131

Преподаватель

Логинов И. П.

Санкт-Петербург, 2023 г.

Задание

Провести рефакторинг приложения, в качестве приложения была взята парная лабораторная работа по дисциплине БЛПС, приведенная к 3-ему этапу (так как 4-й этап полностью меняет логику предыдущих этапов и не содержит сложной логики).
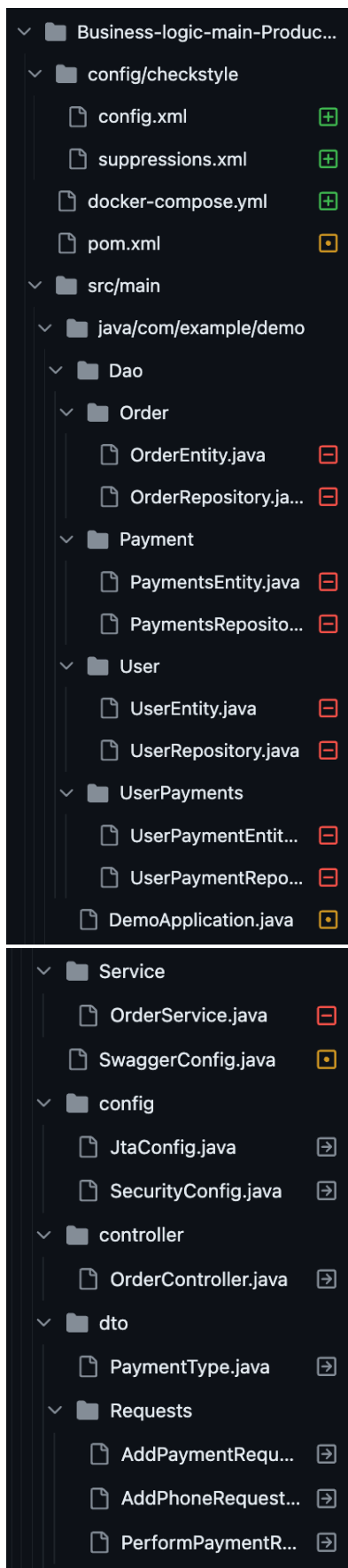
Выполнение работы

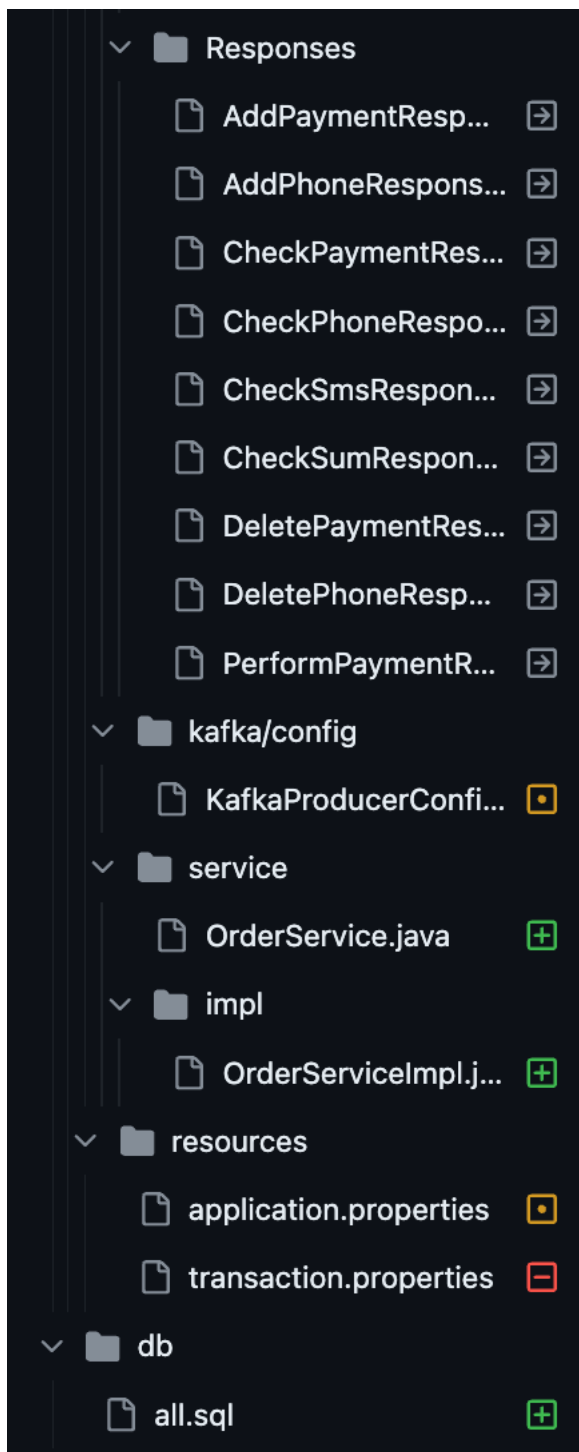В рамках первого этапа были переработан первый компонент приложения Producer:

1. Добавлены линтер checkstyle, openrewrite:
   checkstyle использует google-checks config и suppression
   openrewrite исправляет автоматически ошибки checkstyle а также содержит «рецепты»:

```
<recipe>org.openrewrite.staticanalysis.CodeCleanup</recipe>
<recipe>org.openrewrite.java.OrderImports</recipe>
<recipe>org.openrewrite.staticanalysis.CommonStaticAnalysis</recipe>
<recipe>org.openrewrite.java.format.AutoFormat</recipe>
<recipe>org.openrewrite.java.format.TabsAndIndents</recipe>
<recipe>org.openrewrite.java.format.BlankLines</recipe>
```

2. Убраны лишние фрагменты кода, неиспользуемые модели, репозитории, закомментированные блоки и т. д.
3. Добавили docker-compose.yml для поднятия Postgres БД для сервиса
4. Разнесли сервисы на интерфейсы и реализацию, упростили методы.
5. Исправили нейминг

   Ссылка на коммит: https://github.com/Icerzack/blps-ref/commit/67abb8fb42d8e2cc983d8939cccd57c806d0cced

```
Business-logic-main-Produc...
  config/checkstyle
    config.xml
    suppressions.xml
  docker-compose.yml
  pom.xml
  src/main
    java/com/example/demo
      Dao
        Order
          OrderEntity.java
          OrderRepository.ja...
        Payment
          PaymentsEntity.java
          PaymentsReposito...
        User
          UserEntity.java
          UserRepository.java
        UserPayments
          UserPaymentEntit...
          UserPaymentRepo...
      DemoApplication.java
      Service
        OrderService.java
        SwaggerConfig.java
      config
        JtaConfig.java
        SecurityConfig.java
      controller
        OrderController.java
      dto
        PaymentType.java
        Requests
          AddPaymentRequ...
          AddPhoneRequest...
          PerformPaymentR...
```

```
∨ 📁 Responses
    📄 AddPaymentResp...        ➡
    📄 AddPhoneRespons...      ➡
    📄 CheckPaymentRes...      ➡
    📄 CheckPhoneRespo...      ➡
    📄 CheckSmsRespon...       ➡
    📄 CheckSumRespon...       ➡
    📄 DeletePaymentRes...     ➡
    📄 DeletePhoneResp...      ➡
    📄 PerformPaymentR...      ➡
∨ 📁 kafka/config
    📄 KafkaProducerConfi...   🔸
∨ 📁 service
    📄 OrderService.java       ➕
    ∨ 📁 impl
        📄 OrderServiceImpl.j... ➕
∨ 📁 resources
    📄 application.properties   🔸
    📄 transaction.properties   ➖
∨ 📁 db
    📄 all.sql                 ➕
```

∨ 197 ▪▪▪▪▪ Business-logic-main-Producer/demo/config/checkstyle/suppressions.xml ⎘ ⊞ ...

```
@@ -0,0 +1,197 @@
1    <?xml version="1.0"?>
2
3    <!DOCTYPE suppressions PUBLIC
4        "-//Checkstyle//DTD SuppressionFilter Configuration 1.2//EN"
5        "https://checkstyle.org/dtds/suppressions_1_2.dtd">
6
7    <suppressions>
8      <!-- don't validate generated files -->
9      <suppress id="lineLengthXml" files="releasenotes_old_8\-0_8\-34\.xml"/>
10     <suppress id="lineLengthXml" files="releasenotes_old_6\-0_7\-8\.xml"/>
11     <suppress id="lineLengthXml" files="releasenotes_old_1\-0_5\-9\.xml"/>
12     <suppress id="lineLengthXml" files="[\\/]meta[\\/]"/>
13
14     <suppress checks="FileLength"
15             files="TokenTypes.java|IndentationCheckTest.java|XdocsPagesTest"
16             lines="1"/>
17
18     <!-- illegal words are part of Javadoc -->
19     <suppress checks="TodoComment" files=".*TodoCommentCheck\.java"/>
20
21     <!-- 'Abstract' pattern is used to show it is checking for abstract class name -->
22     <suppress checks="AbstractClassNameCheck"
23             files="AbstractClassNameCheck.java"/>
24     <!-- test should be named as their main class -->
25     <suppress checks="AbstractClassNameCheck"
26             files="AbstractCheckTest.java|AbstractClassNameCheckTest.java|
27                    |AbstractJavadocCheckTest.java|
28                    |AbstractViolationReporterTest.java|AbstractFileSetCheckTest.java|
29                    |AbstractAutomaticBeanTest.java|AbstractClassNameCheckExamplesTest.java"/>
30
31     <!-- Tone down the checking for test code -->
```

∨ 12 ▪▪▪▪▪ Business-logic-main-Producer/demo/docker-compose.yml ⎘ ⊞ ...

```
@@ -0,0 +1,12 @@
1    services:
2      postgres:
3        image: postgres:13
4        environment:
5          POSTGRES_DB: postgres
6          POSTGRES_USER: postgres
7          POSTGRES_PASSWORD: postgres
8        volumes:
9          - ../../db/all.sql:/docker-entrypoint-initdb.d/init.sql
10       ports:
11  +      - "5432:5432"
12       container_name: blps-db
    ⊖
```

∨ ⇳ 72 ▪▪▪▪▪ Business-logic-main-Producer/demo/pom.xml ⎘ ⟨⟩ ▯ ...

```
@@ -16,6 +16,11 @@
16  16      <description>demo</description>
17  17      <properties>
18  18          <java.version>1.8</java.version>
    19              <checkstyle-maven-plugin.version>3.1.2</checkstyle-maven-plugin.version>
    20              <checkstyle.version>8.41</checkstyle.version>
    21              <checkstyle.config.location>config/checkstyle/config.xml</checkstyle.config.location>
    22              <checkstyle.suppressions.location>config/checkstyle/suppressions.xml</checkstyle.suppressions.location>
    23
19  24      </properties>
20  25      <dependencies>
21  26          <dependency>
```

```
@@ -115,6 +120,7 @@
115 120              <groupId>org.apache.kafka</groupId>
116 121              <artifactId>kafka-clients</artifactId>
117 122          </dependency>
    123
118 124      </dependencies>
119 125
120 126      <build>
```

```
@@ -124,7 +130,6 @@
124 130              <artifactId>spring-boot-maven-plugin</artifactId>
125 131          </plugin>
126 132          <plugin>
127                 <!-- Build an executable JAR -->
```

```xml
154              <groupId>org.openrewrite.maven</groupId>
155              <artifactId>rewrite-maven-plugin</artifactId>
156              <version>5.17.1</version>
157              <configuration>
158                  <activeRecipes>
159                      <recipe>org.openrewrite.staticanalysis.CodeCleanup</recipe>
160                      <recipe>org.openrewrite.java.OrderImports</recipe>
161                      <recipe>org.openrewrite.staticanalysis.CommonStaticAnalysis</recipe>                </activeRecipes>
162              </configuration>
163              <executions>
164                  <execution>
165                      <id>validate</id>
166                      <phase>validate</phase>
167                      <goals>
168                          <goal>run</goal>
169                      </goals>
170                  </execution>
171              </executions>
172              <dependencies>
173                  <dependency>
174                      <groupId>org.openrewrite.recipe</groupId>
175                      <artifactId>rewrite-static-analysis</artifactId>
176                      <version>1.1.4</version>
177                  </dependency>
178
179              </dependencies>
180          </plugin>
181          <plugin>
182              <groupId>org.apache.maven.plugins</groupId>
183              <artifactId>maven-checkstyle-plugin</artifactId>
184              <version>3.1.2</version>
185              <configuration>
186                  <configLocation>config/checkstyle/config.xml</configLocation>
187              </configuration>
188              <executions>
189                  <execution>
190                      <id>validate</id>
191                      <phase>validate</phase>
192                      <goals>
193                          <goal>check</goal>
194                      </goals>
195                  </execution>
196              </executions>
```

```xml
150
200    <reporting>
201        <plugins>
202            <plugin>
203                <groupId>org.apache.maven.plugins</groupId>
204                <artifactId>maven-checkstyle-plugin</artifactId>
205                <version>${checkstyle-maven-plugin.version}</version>
206                <reportSets>
207                    <reportSet>
208                        <reports>
209 +                          <report>checkstyle</report>
210                        </reports>
211                        <configuration>
212                            <includeTestSourceDirectory>true</includeTestSourceDirectory>
213                        </configuration>
214                    </reportSet>
215                </reportSets>
216            </plugin>
217        </plugins>
218    </reporting>
151 219  </project>
```

```java
... ...   @@ -1,38 +1,15 @@
  1  1    package com.example.demo;
  2  2
  3        import com.example.demo.Dto.Requests.PerformPaymentRequest;
  4        import org.springframework.boot.CommandLineRunner;
  5  3    import org.springframework.boot.SpringApplication;
  6  4    import org.springframework.boot.autoconfigure.SpringBootApplication;
  7        import org.springframework.context.annotation.Bean;
  8        import org.springframework.http.ResponseEntity;
  9        import org.springframework.kafka.core.KafkaTemplate;
 10  5    import org.springframework.scheduling.annotation.EnableScheduling;
 11        import org.springframework.web.bind.annotation.GetMapping;
 12        import org.springframework.web.bind.annotation.RequestBody;
 13        import org.springframework.web.bind.annotation.RestController;
 14
 15        import java.util.Map;
 16  6
 17  7    @SpringBootApplication
 18        @RestController
 19  8    @EnableScheduling
 20  9    public class DemoApplication {
 21  10
 22         public static void main(String[] args) {
 23             SpringApplication.run(DemoApplication.class, args);
 24         }
 25
 26         @GetMapping("/")
 27         public String allWork() {
 28             return "All works";
 29         }
     11    public static void main(String[] args) {
     12        SpringApplication.run(DemoApplication.class, args);
     13    }
 30  14
 31         //    @Bean
 32         //    CommandLineRunner commandLineRunner(KafkaTemplate<String, PerformPaymentRequest> kafkaTemplate){
 33         //        PerformPaymentRequest request = new PerformPaymentRequest(123456789, "1234567890123456", "12/23", "123", 100.0, "Some address");
 34         //        return args -> {
 35         //            kafkaTemplate.send("perform-test3", request);
 36         //        };
 37         //    }
 38  15    }
```

```
     @@ -14,21 +14,21 @@
14  14   @EnableSwagger2
15  15   public class SwaggerConfig {
16  16
17         @Bean
18         public Docket api() {
19             return new Docket(DocumentationType.SWAGGER_2)
20                 .select()
21                 .apis(RequestHandlerSelectors.any())
22                 .paths(PathSelectors.any())
23                 .build()
24                 .apiInfo(apiInfo());
25         }
    17     @Bean
    18     public Docket api() {
    19         return new Docket(DocumentationType.SWAGGER_2)
    20             .select()
    21             .apis(RequestHandlerSelectors.any())
    22             .paths(PathSelectors.any())
    23             .build()
    24             .apiInfo(apiInfo());
    25     }
26  26
27         private ApiInfo apiInfo() {
28             return new ApiInfoBuilder()
29                 .title("API оформления заказа Delivery Club")
30                 .description("Оформление заказа Delivery Club")
31                 .version("1.0.0")
32                 .build();
33         }
    27     private ApiInfo apiInfo() {
    28         return new ApiInfoBuilder()
    29             .title("API оформления заказа Delivery Club")
    30             .description("Оформление заказа Delivery Club")
    31             .version("1.0.0")
    32             .build();
    33     }
34  34   }
```

```
     @@ -1,4 +1,4 @@
1        package com.example.demo.Config;
     1   package com.example.demo.config;
2    2
3    3   import com.atomikos.icatch.jta.UserTransactionImp;
4    4   import com.atomikos.icatch.jta.UserTransactionManager;
```

```
     @@ -1,4 +1,4 @@
1        package com.example.demo.Config;
     1   package com.example.demo.config;
2    2
3    3   import org.springframework.beans.factory.annotation.Autowired;
4    4   import org.springframework.context.annotation.Bean;
     @@ -30,11 +30,11 @@ public void configure(AuthenticationManagerBuilder auth) throws Exception {
30  30           auth.jdbcAuthentication()
31  31               .dataSource(dataSource)
32  32               .usersByUsernameQuery(
33                       "select phone_number, password, 'true' from users "
34                       "where phone_number=?")
    33                   "select phone_number, password, 'true' from users "
    34                   "where phone_number=?")
35  35               .authoritiesByUsernameQuery(
36                       "select phone_number, role from users "
37                       "where phone_number=?");
    36                   "select phone_number, role from users "
    37                   "where phone_number=?");
38  38           }
39  39
40  40       @Override
     @@ -47,4 +47,3 @@ protected void configure(HttpSecurity http) throws Exception {
47  47               .and().csrf().disable();
48  48           }
49  49   }
50
```

```
@@ -1,10 +1,10 @@
1        package com.example.demo.Controller;
    1    package com.example.demo.controller;
2   2

3        import com.example.demo.Dto.Requests.PerformPaymentRequest;
4        import com.example.demo.Dto.Responses.CheckSmsResponse;
5        import com.example.demo.Dto.Responses.CheckSumResponse;
6        import com.example.demo.Dto.Responses.PerformPaymentResponse;
7        import com.example.demo.Service.OrderService;
    3    import com.example.demo.dto.Requests.PerformPaymentRequest;
    4    import com.example.demo.dto.Responses.CheckSmsResponse;
    5    import com.example.demo.dto.Responses.CheckSumResponse;
    6    import com.example.demo.dto.Responses.PerformPaymentResponse;
    7    import com.example.demo.service.impl.OrderServiceImpl;
8   8    import io.swagger.annotations.Api;
9   9    import io.swagger.annotations.ApiOperation;
10  10   import org.springframework.beans.factory.annotation.Autowired;
@@ -15,18 +15,18 @@
15  15   @Api(tags = "Order", description = "Эндпоинты для взаимодействия с заказом")
16  16   public class OrderController {
17  17       @Autowired
18           private OrderService orderService;
    18       private OrderServiceImpl orderServiceImpl;
19  19

20  20       @GetMapping("/v1/order-management/check-sum")
21  21       @ApiOperation(value = "Проверяет, что сумма больше 100")
22  22       public ResponseEntity<CheckSumResponse> checkSum(@RequestParam int id, @RequestParam int sum) {
23               return orderService.checkSum((long) id, sum);
    23           return orderServiceImpl.checkSum((long) id, sum);
24  24       }
25  25

26  26       @GetMapping("/v1/order-management/check-sms")
27  27       @ApiOperation(value = "Проверяет, что введенный код смс верен для данного номера")
28  28       public ResponseEntity<CheckSmsResponse> checkSms(@RequestParam int id, @RequestParam String phone, @RequestParam String sms) {
29               return orderService.checkSms(id, phone, sms);
    29           return orderServiceImpl.checkSms(id, phone, sms);
30  30       }
31  31

32  32       @PostMapping("/v1/order-management/perform-payment")
```

```
@@ -38,7 +38,6 @@ public ResponseEntity<PerformPaymentResponse> performPayment(@RequestBody Perfor
14           public PerformPaymentRequest(){
    14       public PerformPaymentRequest(){
15  15

16  16       }
17           public PerformPaymentRequest(long userId, String cardNum, String cardDate, String cardCVV, Double cost, String address){
    17       public PerformPaymentRequest(long userId, String cardNum, String cardDate, String cardCVV, Double cost, String address){
18  18           this.userId = userId;
19  19           this.cardNum = cardNum;
20  20           this.cardDate = cardDate;
21  21           this.cardCVV = cardCVV;
22  22           this.cost = cost;
23  23           this.address = address;
24  24       }
25           }
    25       }
```

```
@@ -1,8 +1,8 @@
1        package com.example.demo.Dto.Responses;
    1    package com.example.demo.dto.Responses;
2   2

3   3    import lombok.Data;
4   4

5   5    @Data
6   6    public class AddPaymentResponse {
7   7        private boolean result;
8            }
    8    }
```

```
@@ -1,4 +1,4 @@
1        package com.example.demo.Dto;
    1    package com.example.demo.dto;
2   2

3   3    public enum PaymentType {
4   4        card,
```

```
@@ -1,4 +1,4 @@
1        package com.example.demo.Dto.Requests;
    1    package com.example.demo.dto.Requests;
2   2

3   3    import lombok.Data;
4   4
@@ -8,4 +8,4 @@ public class AddPaymentRequest {
8   8        private String cardNum;
9   9        private String cardDate;
10  10       private String cardCVV;
11           }
    11   }
```

```
∨ 4 ▭▭▭ ...emo/Dto/Responses/AddPaymentResponse.java → ...emo/dto/Responses/AddPaymentResponse.java  ⧉                          ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class AddPaymentResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

```
∨ 4 ▭▭▭▭ .../demo/Dto/Responses/AddPhoneResponse.java → .../demo/dto/Responses/AddPhoneResponse.java  ⧉                         ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class AddPhoneResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

```
∨ ⊹ 4 ▭▭▭▭ ...o/Dto/Responses/CheckPaymentResponse.java → ...o/dto/Responses/CheckPaymentResponse.java  ⧉                      ...
··· ···      @@ -1,4 +1,4 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
 ⊹            @@ -8,4 +8,4 @@ public class CheckPaymentResponse {
  8       8      private String cardNumber;
  9       9      private String cardDate;
 10      10      private String cardCVV;
 11          }
             ⊖
         11  }
```

```
∨ 4 ▭▭▭▭ ...emo/Dto/Responses/CheckPhoneResponse.java → ...emo/dto/Responses/CheckPhoneResponse.java  ⧉                       ...
··· ···      @@ -1,9 +1,9 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class CheckPhoneResponse {
  7       7      private String phone;
  8       8      private boolean result;
  9          }
             ⊖
          9  }
```

```
∨ 4 ▭▭▭▭ .../demo/Dto/Responses/CheckSmsResponse.java → .../demo/dto/Responses/CheckSmsResponse.java  ⧉                       ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class CheckSmsResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

```
∨ 4 ▭▭▭▭ .../demo/Dto/Responses/CheckSumResponse.java → .../demo/dto/Responses/CheckSumResponse.java  ⧉                       ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class CheckSumResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

```
∨ 4 ▭▭▭▭ .../Dto/Responses/DeletePaymentResponse.java → .../dto/Responses/DeletePaymentResponse.java  ⧉                       ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class DeletePaymentResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

```
∨ 4 ▭▭▭▭ ...Dto/Responses/PerformPaymentResponse.java → ...dto/Responses/PerformPaymentResponse.java  ⧉                       ...
··· ···      @@ -1,8 +1,8 @@
  1          package com.example.demo.Dto.Responses;
          1  package com.example.demo.dto.Responses;
  2       2
  3       3  import lombok.Data;
  4       4
  5       5  @Data
  6       6  public class PerformPaymentResponse {
  7       7      private boolean result;
  8          }
             ⊖
          8  }
```

9

```
          @@ -1,6 +1,6 @@
1     1   package com.example.demo.kafka.config;
2     2
3         import com.example.demo.Dto.Requests.PerformPaymentRequest;
      3   import com.example.demo.dto.Requests.PerformPaymentRequest;
4     4   import org.apache.kafka.clients.producer.ProducerConfig;
5     5   import org.apache.kafka.common.serialization.StringSerializer;
6     6   import org.springframework.beans.factory.annotation.Value;
          @@ -19,7 +19,7 @@ public class KafkaProducerConfig {
19    19      @Value("${spring.kafka.bootstrap-servers}")
20    20      private String boostrapServers;
21    21
22        public Map<String, Object> producerConfig(){
      22      public Map<String, Object> producerConfig() {
23    23      HashMap<String, Object> props = new HashMap<>();
24    24      props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, boostrapServers);
25    25      props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);
          @@ -28,12 +28,12 @@ public Map<String, Object> producerConfig(){
28    28      }
29    29
30    30      @Bean
31        public ProducerFactory<String, PerformPaymentRequest> producerFactory(){
      31      public ProducerFactory<String, PerformPaymentRequest> producerFactory(){
32    32          return new DefaultKafkaProducerFactory<>(producerConfig());
33    33      }
34    34
35    35      @Bean
36        public KafkaTemplate<String, PerformPaymentRequest> kafkaTemplate(ProducerFactory<String, PerformPaymentRequest> producerFactory){
      36      public KafkaTemplate<String, PerformPaymentRequest> kafkaTemplate(ProducerFactory<String, PerformPaymentRequest> producerFactory){
37    37          return new KafkaTemplate<>(producerFactory);
38    38      }
39    39   }
```

```
          @@ -0,0 +1,14 @@
      1   package com.example.demo.service;
      2
      3   import com.example.demo.dto.Responses.CheckSmsResponse;
      4   import com.example.demo.dto.Responses.CheckSumResponse;
      5   import com.example.demo.dto.Responses.PerformPaymentResponse;
      6   import org.springframework.http.ResponseEntity;
      7
      8   public interface OrderService {
      9      ResponseEntity<CheckSumResponse> checkSum(Long orderId, double sumThreshold);
      10
      11     ResponseEntity<CheckSmsResponse> checkSms(Integer orderId, String phoneNumber, String sms);
      12
      13     ResponseEntity<PerformPaymentResponse> performPayment(long userId, String cardNum, String cardDate, String cardCVV, Double cost, String address);
      14  }
```

```
          @@ -0,0 +1,69 @@
      1   package com.example.demo.service.impl;
      2
      3   import com.example.demo.dto.Requests.PerformPaymentRequest;
      4   import com.example.demo.dto.Responses.CheckSmsResponse;
      5   import com.example.demo.dto.Responses.CheckSumResponse;
      6   import com.example.demo.dto.Responses.PerformPaymentResponse;
      7   import com.example.demo.service.OrderService;
      8   import org.apache.commons.codec.digest.DigestUtils;
      9   import org.apache.commons.lang3.StringUtils;
      10  import org.springframework.http.ResponseEntity;
      11  import org.springframework.kafka.core.KafkaTemplate;
      12  import org.springframework.stereotype.Service;
      13
      14  @Service
      15  public class OrderServiceImpl implements OrderService {
      16     private KafkaTemplate<String, PerformPaymentRequest> kafkaTemplate;
      17
      18     public OrderServiceImpl(KafkaTemplate<String, PerformPaymentRequest> kafkaTemplate) {
      19         this.kafkaTemplate = kafkaTemplate;
      20     }
      21
      22     public ResponseEntity<CheckSumResponse> checkSum(Long orderId, double sumThreshold) {
      23         CheckSumResponse checkSumResponse = new CheckSumResponse();
      24         boolean isSumValid = sumValidation(sumThreshold);
      25         checkSumResponse.setResult(isSumValid);
      26         return ResponseEntity.ok(checkSumResponse);
      27     }
      28
      29     private boolean sumValidation(double sum) {
      30         return sum > 100;
      31     }
      32
      33     public ResponseEntity<CheckSmsResponse> checkSms(Integer orderId, String phoneNumber,
      34                                                      String sms) {
      35         CheckSmsResponse checkSmsResponse = new CheckSmsResponse();
      36         boolean isSmsValid = validateSms(orderId, phoneNumber, sms);
      37         checkSmsResponse.setResult(isSmsValid);
      38         return ResponseEntity.ok(checkSmsResponse);
      39     }
      40
      41     private boolean validateSms(Integer orderId, String phoneNumber, String sms) {
      42         String sha256sms = DigestUtils.sha256Hex(orderId + phoneNumber).replaceAll("[^0-9]", "");
      43         String lastFourDigits = StringUtils.right(sha256sms, 4);
      44         return sms.equals(lastFourDigits);
      45     }
      46
      47     public ResponseEntity<PerformPaymentResponse> performPayment(long userId, String cardNum,
      48                                                                  String cardDate, String cardCVV,
      49                                                                  Double cost, String address) {
      50         PerformPaymentResponse performPaymentResponse = new PerformPaymentResponse();
      51         boolean isValidPayment = validatePaymentDetails(cardNum, cardCVV);
      52
      53         if (!isValidPayment) {
      54             performPaymentResponse.setResult(false);
      55             return ResponseEntity.ok(performPaymentResponse);
      56         }
      57
      58         PerformPaymentRequest performPaymentRequest =
      59                 new PerformPaymentRequest(userId, cardNum, cardDate, cardCVV, cost, address);
      60         kafkaTemplate.send("perform-test3", performPaymentRequest);
      61         performPaymentResponse.setResult(true);
      62         return ResponseEntity.ok(performPaymentResponse);
      63     }
      64
      65     private boolean validatePaymentDetails(String cardNum, String cardCVV) {
      66         return cardNum.matches("[-+]?\\d+") && cardNum.length() >= 13 && cardNum.length() <= 19
      67             && cardCVV.matches("[-+]?\\d+") && cardCVV.length() == 3;
      68     }
      69  }
```