

Case Study 2: Java-Based Configuration

Case Study Title: E-Commerce Order Processing

POJO Classes:

1. Product.java
 - addProduct(), listProducts()
2. Order.java
 - createOrder(), cancelOrder()
3. Payment.java
 - processPayment(), refundPayment()

Code:

//Product.java

package com.ecommerce;

import java.util.ArrayList;

import java.util.List;

public class Product {

private List<String> productList = **new** ArrayList<>();

public void addProduct(String product) {

 productList.add(product);

 System.**out**.println("Product added: " + product);

 }

public void listProducts() {

 System.**out**.println("Available products:");

 productList.forEach(System.**out**::println);

 }

}

//Order.java

package com.ecommerce;

public class Order {

public void createOrder(String product) {

 System.**out**.println("Order created for: " + product);

 }

public void cancelOrder(String product) {

 System.**out**.println("Order cancelled for: " + product);

 }

}

```
//Payment.java
```

```
package com.ecommerce;
```

```
public class Payment {
```

```
    public void processPayment(String product) {  
        System.out.println("Payment processed for: " + product);  
    }
```

```
    public void refundPayment(String product) {  
        System.out.println("Payment refunded for: " + product);  
    }
```

```
}
```

```
//EcommerceService.java
```

```
package com.ecommerce;
```

```
public class EcommerceService {
```

```
    private Product product;  
    private Order order;  
    private Payment payment;
```

```
    public EcommerceService(Product product, Order order, Payment payment) {  
        this.product = product;  
        this.order = order;  
        this.payment = payment;  
    }
```

```
    public void placeOrder(String productName) {  
        product.addProduct(productName);  
        order.createOrder(productName);  
        payment.processPayment(productName);  
    }
```

```
    public void cancelOrder(String productName) {  
        order.cancelOrder(productName);  
        payment.refundPayment(productName);  
    }
```

```
    public void showProducts() {  
        product.listProducts();  
    }
```

```
}
```

```
//AppConfig.java
```

```
package com.ecommerce;
```

```

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
@Configuration
public class AppConfig {

    @Bean
    public Product product() {
        return new Product();
    }

    @Bean
    public Order order() {
        return new Order();
    }

    @Bean
    public Payment payment() {
        return new Payment();
    }

    @Bean
    public EcommerceService ecommerceService() {
        return new EcommerceService(product(), order(), payment());
    }

}

//Main.java
package com.ecommerce;

import
org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class Main {

    public static void main(String[] args) {
        AnnotationConfigApplicationContext context =
            new AnnotationConfigApplicationContext(AppConfig.class);

        EcommerceService service = context.getBean(EcommerceService.class);

        service.placeOrder("Laptop");
        service.placeOrder("Smartphone");
        service.showProducts();
    }
}

```

```
        service.cancelOrder("Laptop");

        context.close();
    }

}
```

Output: Laptop

Smartphone

Order cancelled for: Laptop

Payment refunded for: Laptop