

## Microservices Workflow (E-Commerce Example)

**Design and implement a microservice-based order processing workflow where a customer purchases two laptops. The system should validate product availability, calculate the total amount, process the payment, update the order status, reduce stock, and return a confirmation response to the customer.**

**// EUREKA SERVER**

**// ClassAssessmentEurekaServerApplication.java**

```
package com.example.eurekaservice;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;
```

```
@SpringBootApplication
```

```
@EnableEurekaServer
```

```
public class ClassAssessmentEurekaServerApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(ClassAssessmentEurekaServerApplication.class, args);
```

```
    }
```

```
}
```

**// application.properties**

```
spring.application.name=class_assessment-eureka-server
```

```
server.port=8761
```

```
eureka.client.register-with-eureka=false
```

eureka.client.fetch-registry=false

## **// PRODUCT SERVICE**

### **// ClassAssessmentProductServiceApplication.java**

```
package com.example.productservice;
```

```
import org.springframework.boot.SpringApplication;
```

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

```
@SpringBootApplication
```

```
public class ClassAssessmentProductServiceApplication {
```

```
    public static void main(String[] args) {
```

```
        SpringApplication.run(ClassAssessmentProductServiceApplication.class, args);
```

```
    }
```

```
}
```

### **// Product.java**

```
package com.example.productservice.entity;
```

```
public class Product {
```

```
    private String id;
```

```
    private String name;
```

```
private double price;

private int stock;


public Product(String id, String name, double price, int stock) {

    this.id = id;

    this.name = name;

    this.price = price;

    this.stock = stock;

}

public String getId() { return id; }

public void setId(String id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public double getPrice() { return price; }

public void setPrice(double price) { this.price = price; }

public int getStock() { return stock; }

public void setStock(int stock) { this.stock = stock; }

}
```

#### **// ProductController.java**

```
package com.example.productservice.controller;


import java.util.*;

import java.util.concurrent.ConcurrentHashMap;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import com.example.productservice.entity.Product;
```

```
import jakarta.annotation.PostConstruct;
```

```
@RestController
```

```
@RequestMapping("/products")
```

```
public class ProductController {
```

```
    private Map<String, Product> productMap = new ConcurrentHashMap<>();
```

```
@PostConstruct
```

```
public void init() {
```

```
    productMap.put("P101", new Product("P101", "Laptop", 1200, 10));
```

```
}
```

```
@GetMapping
```

```
public Collection<Product> getAllProducts() {
```

```
    return productMap.values();
```

```
}
```

```
@GetMapping("/{id}")
```

```
public Product getProductById(@PathVariable String id) {
```

```
    return productMap.get(id);
```

```
}
```

```

    @PutMapping("/{id}/reduceStock")

    public String reduceStock(@PathVariable String id, @RequestParam int qty) {

        Product p = productMap.get(id);

        if (p != null && p.getStock() >= qty) {

            p.setStock(p.getStock() - qty);

            return "Stock updated";

        }

        return "Insufficient stock";

    }

}

```

#### **// application.properties**

```

spring.application.name=class_assessment-product-service

server.port=8081

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

```

#### **// PAYMENT SERVICE**

##### **// ClassAssessmentPaymentServiceApplication.java**

```

package com.example.paymentservice;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

```

@SpringBootApplication

@EnableDiscoveryClient

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

**// Payment.java**

package com.example.paymentservice.entity;

```
public class Payment {  
  
    private String orderId;  
  
    private double amount;  
  
    private String paymentMethod;  
  
    public String getOrderId() { return orderId; }  
  
    public void setOrderId(String orderId) { this.orderId = orderId; }  
  
    public double getAmount() { return amount; }  
  
    public void setAmount(double amount) { this.amount = amount; }  
  
    public String getPaymentMethod() { return paymentMethod; }  
  
    public void setPaymentMethod(String paymentMethod) { this.paymentMethod =  
paymentMethod; }  
  
}
```

### **// PaymentController.java**

```
package com.example.paymentservice.controller;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import com.example.paymentservice.entity.Payment;
```

```
@RestController
```

```
@RequestMapping("/payments")
```

```
public class PaymentController {
```

```
    @PostMapping
```

```
    public String processPayment(@RequestBody Payment payment) {
```

```
        return "SUCCESS";
```

```
    }
```

```
}
```

### **// application.properties**

```
spring.application.name=class_assessment-payment-service
```

```
server.port=8083
```

```
eureka.client.service-url.defaultZone=http://localhost:8761/eureka
```

### **// ORDER SERVICE (8082)**

#### **// ClassAssessmentOrderServiceApplication.java**

```
package com.example.orderservice;
```

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

import org.springframework.cloud.openfeign.EnableFeignClients;


@SpringBootApplication

@EnableDiscoveryClient

@EnableFeignClients

public class ClassAssessmentOrderServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(ClassAssessmentOrderServiceApplication.class, args);

    }

}
```

#### **// Order.java**

```
package com.example.orderservice.entity;


public class Order {

    private String orderId;

    private String productId;

    private int quantity;

    private double totalAmount;

    private String status;

    public String getOrderId() { return orderId; }
```



```
public void setOrderId(String orderId) { this.orderId = orderId; }

public String getProductId() { return productId; }

public void setProductId(String productId) { this.productId = productId; }

public int getQuantity() { return quantity; }

public void setQuantity(int quantity) { this.quantity = quantity; }

public double getTotalAmount() { return totalAmount; }

public void setTotalAmount(double totalAmount) { this.totalAmount = totalAmount; }

public String getStatus() { return status; }

public void setStatus(String status) { this.status = status; }

}
```

#### **// Product.java (DTO)**

```
package com.example.orderservice.Dto;

public class Product {

    private String id;

    private String name;

    private double price;

    private int stock;

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }

    public String getName() { return name; }

    public void setName(String name) { this.name = name; }

    public double getPrice() { return price; }
```

```
public void setPrice(double price) { this.price = price; }

public int getStock() { return stock; }

public void setStock(int stock) { this.stock = stock; }

}
```

#### **// PaymentRequest.java (DTO)**

```
package com.example.orderservice.Dto;
```

```
public class PaymentRequest {

    private String orderId;

    private double amount;

    private String paymentMethod;

    public PaymentRequest(String orderId, double amount, String paymentMethod) {

        this.orderId = orderId;

        this.amount = amount;

        this.paymentMethod = paymentMethod;

    }

    public String getOrderId() { return orderId; }

    public void setOrderId(String orderId) { this.orderId = orderId; }

    public double getAmount() { return amount; }

    public void setAmount(double amount) { this.amount = amount; }

    public String getPaymentMethod() { return paymentMethod; }

    public void setPaymentMethod(String paymentMethod) { this.paymentMethod = paymentMethod; }

}
```

```
}
```

### **// ProductClient.java**

```
package com.example.orderservice.feign;
```

```
import org.springframework.cloud.openfeign.FeignClient;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import com.example.orderservice.Dto.Product;
```

```
@FeignClient(name = "product-service")
```

```
public interface ProductClient {
```

```
    @GetMapping("/products/{id}")
```

```
    Product getProduct(@PathVariable("id") String id);
```

```
    @PutMapping("/products/{id}/reduceStock")
```

```
    String reduceStock(@PathVariable("id") String id, @RequestParam int qty);
```

```
}
```

### **// PaymentClient.java**

```
package com.example.orderservice.feign;
```

```
import org.springframework.cloud.openfeign.FeignClient;
```

```
import org.springframework.web.bind.annotation.*;
```

```
import com.example.orderservice.Dto.PaymentRequest;
```

```
@FeignClient(name = "payment-service")

public interface PaymentClient {

    @PostMapping("/payments")

    String pay(@RequestBody PaymentRequest request);

}
```

#### **// OrderController.java**

```
package com.example.orderservice.controller;

import java.util.*;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import com.example.orderservice.Dto.*;
import com.example.orderservice.entity.Order;
import com.example.orderservice.feign.*;

@RestController

@RequestMapping("/orders")

public class OrderController {

    @Autowired

    private ProductClient productClient;
```

@Autowired

private PaymentClient paymentClient;

private Map<String, Order> orders = new HashMap<>();

@PostMapping

public ResponseEntity<?> placeOrder(@RequestBody Order order) {

    Product p = productClient.getProduct(order.getProductId());

    if (p == null || p.getStock() < order.getQuantity()) {

        return ResponseEntity.badRequest().body("Product not available or insufficient stock");

    }

    double total = p.getPrice() \* order.getQuantity();

    order.setOrderId("O" + new Random().nextInt(10000));

    order.setTotalAmount(total);

    order.setStatus("PENDING\_PAYMENT");

    PaymentRequest payment = new PaymentRequest(order.getOrderId(), total, "Credit Card");

    String paymentStatus = paymentClient.pay(payment);

    if ("SUCCESS".equals(paymentStatus)) {

        order.setStatus("CONFIRMED");

```
        productClient.reduceStock(order.getProductId(), order.getQuantity());  
    }  
}
```

```
orders.put(order.getId(), order);  
return ResponseEntity.ok(order);  
}
```

```
@GetMapping  
public Collection<Order> allOrders() {  
    return orders.values();  
}  
}
```

#### **// application.properties**

spring.application.name=class\_assessment-order-service

server.port=8082

eureka.client.service-url.defaultZone=http://localhost:8761/eureka