

Order Service — pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" ...>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>order-service</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <properties>
    <java.version>17</java.version>
    <spring-boot.version>3.1.4</spring-boot.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <!-- Spring Cloud Sleuth -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-sleuth</artifactId>
    </dependency>
    <!-- Zipkin reporter (brave) -->
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-sleuth-zipkin</artifactId>
    </dependency>
    <!-- For RestTemplate calls -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>

  <dependencyManagement>
    <!-- include spring cloud bom if needed -->
  </dependencyManagement>
</project>
```

Order Service — application.properties

```
spring.application.name=order-service
server.port=8081
```

```
# Zipkin (where traces are reported)
spring.zipkin.base-url=http://localhost:9411/
spring.sleuth.sampler.probability=1.0
```

Order Service — OrderController.java

```
package com.example.orderservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.RestTemplate;
import org.springframework.http.ResponseEntity;
import org.springframework.context.annotation.Bean;
import @SpringBootApplication
```

```

@RestController
public class OrderApplication {

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

    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }

    @PostMapping("/orders")
    public ResponseEntity<String> createOrder(@RequestBody String orderPayload) {
        // Log start (Sleuth will attach trace IDs automatically to logs)
        System.out.println("OrderService: Received order payload: " + orderPayload);

        // Call payment service
        String paymentUrl = "http://localhost:8082/payments";
        RestTemplate rt = restTemplate();
        ResponseEntity<String> resp = rt.postForEntity(paymentUrl, orderPayload, String.class);

        // Return confirmation including payment response
        String confirmation = "Order confirmed. Payment service response: " + resp.getBody();
        System.out.println("OrderService: Returning confirmation");
        return ResponseEntity.ok(confirmation);
    }
}

```

Payment Service — pom.xml

```

<project ...>
<!-- similar to order-service but artifactId=payment-service -->
<groupId>com.example</groupId>
<artifactId>payment-service</artifactId>
<version>0.0.1-SNAPSHOT</version>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-sleuth-zipkin</artifactId>
</dependency>
</dependencies>
</project>

```

Payment Service — application.properties

```

spring.application.name=payment-service
server.port=8082

```

```

spring.zipkin.base-url=http://localhost:9411/
spring.sleuth.sampler.probability=1.0

```

Payment Service — PaymentController.java

```
package com.example.paymentservice;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.*;
import org.springframework.http.ResponseEntity;

@SpringBootApplication
@RestController
public class PaymentApplication {

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

    @PostMapping("/payments")
    public ResponseEntity<String> processPayment(@RequestBody String payload) throws InterruptedException {
        System.out.println("PaymentService: Processing payment for payload: " + payload);
        // Simulate processing delay
        Thread.sleep(120);
        System.out.println("PaymentService: Payment processed");
        return ResponseEntity.ok("payment-success");
    }
}
```

Zipkin — docker-compose.yml

```
version: '3'
services:
  zipkin:
    image: openzipkin/zipkin
    ports:
      - "9411:9411"
```