

Case Study: Order & Payment Microservices with Zipkin Tracing

Project Structure

order-payment-zipkin/

|

|— pom.xml

|

|— order-service/

|

|— pom.xml

|

|— src/main/java/com/example/orderservice/

|

|— OrderServiceApplication.java

|

|— controller/

|

|— OrderController.java

|

|— config/

|

|— RestTemplateConfig.java

|

|— src/main/resources/

|

|— application.properties

|

|— payment-service/

|

|— pom.xml

|

|— src/main/java/com/example/paymentservice/

|

|— PaymentServiceApplication.java

|

|— controller/

|

|— PaymentController.java

|

|— src/main/resources/

```
|   └─ application.properties
|
└─ zipkin/
    └─ zipkin-server-3.x.x-exec.jar  # Downloaded Zipkin server jar
```

Parent pom.xml (order-payment-zipkin/pom.xml)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>order-payment-zipkin</artifactId>
  <version>1.0.0</version>
  <packaging>pom</packaging>

  <modules>
    <module>order-service</module>
    <module>payment-service</module>
  </modules>

  <properties>
    <java.version>17</java.version>
    <spring.boot.version>3.2.5</spring.boot.version>
    <spring.cloud.version>2023.0.1</spring.cloud.version>
  </properties>
```

```

<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-dependencies</artifactId>
      <version>${spring.boot.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>

    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>${spring.cloud.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
</project>

```

order-service/pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

```

```
<parent>
  <groupId>com.example</groupId>
  <artifactId>order-payment-zipkin</artifactId>
  <version>1.0.0</version>
</parent>
```

```
<artifactId>order-service</artifactId>
```

```
<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>
```

OrderServiceApplication.java

```
package com.example.orderservice;

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

@SpringBootApplication
public class OrderServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(OrderServiceApplication.class, args);

    }

}
```

RestTemplateConfig.java

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

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.client.RestTemplate;

@Configuration
public class RestTemplateConfig {

    @Bean
    public RestTemplate restTemplate() {

        return new RestTemplate();

    }

}
```

OrderController.java

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

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.client.RestTemplate;

@RestController
@RequestMapping("/orders")
public class OrderController {

    private static final Logger log = LoggerFactory.getLogger(OrderController.class);

    @Autowired
    private RestTemplate restTemplate;

    @PostMapping
    public String createOrder(@RequestBody String order) {
        log.info("Order Service - Received order: {}", order);
        String paymentResponse = restTemplate.postForObject(
            "http://localhost:8082/payment", order, String.class);
        return "Order placed! Payment status: " + paymentResponse;
    }
}

```

order-service/src/main/resources/application.properties

```

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

```

spring.zipkin.base-url=http://localhost:9411

spring.sleuth.sampler.probability=1.0

payment-service/pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<parent>
```

```
  <groupId>com.example</groupId>
```

```
  <artifactId>order-payment-zipkin</artifactId>
```

```
  <version>1.0.0</version>
```

```
</parent>
```

```
<artifactId>payment-service</artifactId>
```

```
<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>
```

PaymentServiceApplication.java

```
package com.example.paymentservice;

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

@SpringBootApplication

public class PaymentServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(PaymentServiceApplication.class, args);

    }

}
```

PaymentController.java

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

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.web.bind.annotation.*;
```



```
@RestController
@RequestMapping("/payment")
public class PaymentController {

    private static final Logger log = LoggerFactory.getLogger(PaymentController.class);

    @PostMapping
    public String processPayment(@RequestBody String payment) {
        log.info("Payment Service - Processing payment for: {}", payment);
        try {
            Thread.sleep(150); // Simulate delay
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
        log.info("Payment processed successfully for: {}", payment);
        return "Payment successful";
    }
}
```

payment-service/src/main/resources/application.properties

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

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