

Case Study: Order & Payment Microservices with Zipkin Tracing Business

application.properties

spring.application.name=zipkin-payment-service

server.port=8082

management.zipkin.tracing.endpoint=http://localhost:9411/api/v2/spans
management.tracing.sampling.probability=1.0

Payment.java

package com.example.zipkinpaymentservice.model;

public class Payment {

private Long orderId;

private double amount;

public Payment(Long orderId, double amount) {

this.orderId = orderId;

this.amount = amount;

}

public Long getOrderId() {

return orderId;

}

public double getAmount() {

return amount;

}

}

PaymentService

package com.example.zipkinpaymentservice.service;

import com.example.zipkinpaymentservice.model.Payment;

import io.micrometer.tracing.Tracer;

```
import io.micrometer.tracing.Span;
import org.springframework.stereotype.Service;
```

```
import java.util.ArrayList;
import java.util.List;
@Service
public class PaymentService {
    private final Tracer tracer;
    private final List payments = new ArrayList<>();
    public PaymentService(Tracer tracer) {
        this.tracer = tracer;
    }
    public String processPayment(Payment payment) {
        payments.add(payment);
```

```
        Span newSpan = tracer.nextSpan().name("process-payment").start();
        try (Tracer.SpanInScope ws = tracer.withSpan(newSpan)) {
            return "Payment processed for Order ID: " + payment.getId();
        }
        finally {
            newSpan.end();
        }
    }
    public List getPayments() {
        return payments;
    }
}
```

```
PaymentController package
com.example.zipkinpaymentservice.controller;
import com.example.zipkinpaymentservice.model.Payment;
import com.example.zipkinpaymentservice.service.PaymentService;
import org.springframework.web.bind.annotation.*;
import java.util.List;
```

```
@RestController  
public class PaymentController {  
private final PaymentService paymentService;  
public PaymentController(PaymentService paymentService) {  
    this.paymentService = paymentService;  
}  
@PostMapping("/payments")  
public String processPayment(@RequestBody Payment payment) {  
    return paymentService.processPayment(payment);  
}  
@GetMapping("/payments")  
public List getPayments() {  
    return paymentService.getPayments();  
}  
}
```