

Webserviceconfig

```
package com.example.paymentgateway.config;

import org.springframework.boot.web.servlet.ServletRegistrationBean;
import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.io.ClassPathResource;
import org.springframework.ws.config.annotation.EnableWs;
import org.springframework.ws.config.annotation.WsConfigurerAdapter;
import org.springframework.ws.transport.http.MessageDispatcherServlet;
import org.springframework.ws.wsdl.wsdl11.DefaultWsdl11Definition;
import org.springframework.xml.xsd.SimpleXsdSchema;
import org.springframework.xml.xsd.XsdSchema;

@EnableWs
@Configuration
public class WebServiceConfig extends WsConfigurerAdapter {

    @Bean
    public ServletRegistrationBean<MessageDispatcherServlet>
messageDispatcherServlet(ApplicationContext applicationContext) {
        MessageDispatcherServlet servlet = new MessageDispatcherServlet();
        servlet.setApplicationContext(applicationContext);
        servlet.setTransformWsdlLocations(true);
        return new ServletRegistrationBean<>(servlet, "/ws/*");
    }

    @Bean(name = "payments")
    public DefaultWsdl11Definition defaultWsdl11Definition(XsdSchema
paymentsSchema) {
        DefaultWsdl11Definition wsdl11Definition = new
DefaultWsdl11Definition();
        wsdl11Definition.setPortTypeName("PaymentsPort");
        wsdl11Definition.setLocationUri("/ws");

wsdl11Definition.setTargetNamespace("http://example.com/paymentgateway");
        wsdl11Definition.setSchema(paymentsSchema);
        return wsdl11Definition;
    }

    @Bean
    public XsdSchema paymentsSchema() {
        return new SimpleXsdSchema(new ClassPathResource("payments.xsd"));
    }
}
```

restpaymentcontroller

```
package com.example.paymentgateway.controller;

import com.example.paymentgateway.model.Payment;
import com.example.paymentgateway.service.PaymentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

@RestController
@RequestMapping("/api/payments")
public class RestPaymentController {

    @Autowired
    private PaymentService paymentService;

    @PostMapping("/process")
    public ResponseEntity<Payment> processPayment(@RequestBody Payment payment)
    {
        Payment processedPayment = paymentService.processPayment(payment);
        return ResponseEntity.ok(processedPayment);
    }

    @GetMapping("/status/{transactionId}")
    public ResponseEntity<Payment> getPaymentStatus(@PathVariable String
transactionId) {
        Payment payment = paymentService.getPaymentStatus(transactionId);
        if (payment != null) {
            return ResponseEntity.ok(payment);
        }
        return ResponseEntity.notFound().build();
    }

    @PostMapping("/refund/{transactionId}")
    public ResponseEntity<Payment> refundPayment(
        @PathVariable String transactionId,
        @RequestParam Double amount) {
        Payment refundedPayment = paymentService.refundPayment(transactionId,
amount);
        if (refundedPayment != null) {
            return ResponseEntity.ok(refundedPayment);
        }
        return ResponseEntity.badRequest().build();
    }
}
```

paymentendpoint

```
package com.example.paymentgateway.endpoint;

import com.example.paymentgateway.model.Payment;
import com.example.paymentgateway.service.PaymentService;
import com.example.paymentgateway.soap.PaymentRequest;
import com.example.paymentgateway.soap.PaymentResponse;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.ws.server.endpoint.annotation.Endpoint;
import org.springframework.ws.server.endpoint.annotation.PayloadRoot;
import org.springframework.ws.server.endpoint.annotation.RequestPayload;
import org.springframework.ws.server.endpoint.annotation.ResponsePayload;

@Endpoint
public class PaymentEndpoint {
    private static final String NAMESPACE_URI =
"http://example.com/paymentgateway";

    @Autowired
    private PaymentService paymentService;

    @PayloadRoot(namespace = NAMESPACE_URI, localPart = "PaymentRequest")
    @ResponsePayload
    public PaymentResponse processPayment(@RequestPayload PaymentRequest
request) {
        Payment payment = new Payment();
        payment.setAmount(request.getAmount());
        payment.setCurrency(request.getCurrency());
        payment.setCardNumber(request.getCardNumber());
        payment.setCardHolderName(request.getCardHolderName());
        payment.setExpiryDate(request.getExpiryDate());
        payment.setCvv(request.getCvv());

        Payment processedPayment = paymentService.processPayment(payment);

        PaymentResponse response = new PaymentResponse();
        response.setTransactionId(processedPayment.getTransactionId());
        response.setStatus(processedPayment.getStatus());
        response.setDescription(processedPayment.getDescription());
        response.setAmount(processedPayment.getAmount());
        response.setCurrency(processedPayment.getCurrency());

        return response;
    }
}
```

payment

```
package com.example.paymentgateway.model;

import lombok.Data;
import javax.persistence.*;
import java.time.LocalDateTime;

@Data
@Entity
@Table(name = "payments")
public class Payment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String transactionId;
    private Double amount;
    private String currency;
    private String cardNumber;
    private String cardHolderName;
    private String expiryDate;
    private String cvv;
    private String status;
    private String description;
    private LocalDateTime createdAt;
    private LocalDateTime updatedAt;

    @PrePersist
    protected void onCreate() {
        createdAt = LocalDateTime.now();
        updatedAt = LocalDateTime.now();
    }

    @PreUpdate
    protected void onUpdate() {
        updatedAt = LocalDateTime.now();
    }
}
```

paymentrepository

```
package com.example.paymentgateway.repository;

import com.example.paymentgateway.model.Payment;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface PaymentRepository extends JpaRepository<Payment, Long> {
```

```
Payment findByIdByTransactionId(String transactionId);  
}
```

paymentserviceimpl

```
package com.example.paymentgateway.service.impl;  
  
import com.example.paymentgateway.model.Payment;  
import com.example.paymentgateway.repository.PaymentRepository;  
import com.example.paymentgateway.service.PaymentService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.UUID;  
  
@Service  
public class PaymentServiceImpl implements PaymentService {  
  
    @Autowired  
    private PaymentRepository paymentRepository;  
  
    @Override  
    public Payment processPayment(Payment payment) {  
        payment.setTransactionId(UUID.randomUUID().toString());  
        payment.setStatus("PROCESSING");  
  
        // Simulate payment processing  
        try {  
            Thread.sleep(1000); // Simulate processing time  
            payment.setStatus("SUCCESS");  
        } catch (InterruptedException e) {  
            payment.setStatus("FAILED");  
            payment.setDescription("Payment processing failed");  
        }  
  
        return paymentRepository.save(payment);  
    }  
  
    @Override  
    public Payment getPaymentStatus(String transactionId) {  
        return paymentRepository.findByIdByTransactionId(transactionId);  
    }  
  
    @Override  
    public Payment refundPayment(String transactionId, Double amount) {  
        Payment payment = paymentRepository.findByIdByTransactionId(transactionId);  
        if (payment != null && "SUCCESS".equals(payment.getStatus())) {  
            payment.setStatus("REFUNDED");  
            payment.setDescription("Refunded amount: " + amount);  
            return paymentRepository.save(payment);  
        }  
    }  
}
```

```
    }  
    return null;  
}  
}
```

paymentservice

```
package com.example.paymentgateway.service;  
  
import com.example.paymentgateway.model.Payment;  
  
public interface PaymentService {  
    Payment processPayment(Payment payment);  
    Payment getPaymentStatus(String transactionId);  
    Payment refundPayment(String transactionId, Double amount);  
}  
Payment request  
  
package com.example.paymentgateway.soap;  
  
import lombok.Data;  
  
import javax.xml.bind.annotation.XmlAccessType;  
import javax.xml.bind.annotation.XmlAccessorType;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlRootElement;  
  
@Data  
@XmlRootElement(name = "PaymentRequest")  
@XmlAccessorType(XmlAccessType.FIELD)  
public class PaymentRequest {  
    @XmlElement(required = true)  
    private Double amount;  
  
    @XmlElement(required = true)  
    private String currency;  
  
    @XmlElement(required = true)  
    private String cardNumber;  
  
    @XmlElement(required = true)  
    private String cardHolderName;  
  
    @XmlElement(required = true)  
    private String expiryDate;  
  
    @XmlElement(required = true)  
    private String cvv;  
}
```

Payment response

```
package com.example.paymentgateway.soap;

import lombok.Data;

import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@Data
@XmlRootElement(name = "PaymentResponse")
@XmlAccessorType(XmlAccessType.FIELD)
public class PaymentResponse {
    @XmlElement(required = true)
    private String transactionId;

    @XmlElement(required = true)
    private String status;

    @XmlElement
    private String description;

    @XmlElement(required = true)
    private Double amount;

    @XmlElement(required = true)
    private String currency;
}
```

paymetgatewayapplication.java

```
package com.example.paymentgateway;

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

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