Case Study: Order Processing System Using Kafka and Spring Boot

application.yml

```yaml
spring:
 kafka:
  bootstrap.servers: localhost:9092
  producer:
   key-serializer: org.apache.kafka.common.serialization.StringSerializer
   value-serializer: org.springframework.kafka.support.serializer.JsonSerializer
  consumer:
   group-id: order-group
   key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
   value-deserializer: org.springframework.kafka.support.serializer.JsonDeserializer
  properties:
   spring.json.trusted.packages: "*"
```

Order.java

```java
package com.example.orderprocessingsystem.entity;
import java.util.Date;
import com.fasterxml.jackson.annotation.JsonFormat;
public class Order {

	private String orderId;
	private String customerName;
	private String productName;
	private int quantity;
	private double price;

	@JsonFormat(shape = JsonFormat.Shape.STRING, pattern = "yyyy-MM-dd")
	private Date orderDate;

	public Order() {}
	public Order(String orderId, String customerName, String productName, int quantity,
double price, Date orderDate) {
		this.orderId = orderId;
		this.customerName = customerName;
		this.productName = productName;
		this.quantity = quantity;
		this.price = price;
		this.orderDate = orderDate;
	}
	public String getOrderId() {
		return orderId;
```

```
        }
        public void setOrderId(String orderId) {
                this.orderId = orderId;
        }
        public String getCustomerName() {
                return customerName;
        }
        public void setCustomerName(String customerName) {
                this.customerName = customerName;
        }
        public String getProductName() {
                return productName;
        }
        public void setProductName(String productName) {
                this.productName = productName;
        }
        public int getQuantity() {
                return quantity;
        }
        public void setQuantity(int quantity) {
                this.quantity = quantity;
        }
        public double getPrice() {
                return price;
        }
        public void setPrice(double price) {
                this.price = price;
        }
        public Date getOrderDate() {
                return orderDate;
        }
        public void setOrderDate(Date orderDate) {
                this.orderDate = orderDate;
        }

        @Override
  public String toString() {
     return "Order(orderId=" + orderId +
          ", customerName=" + customerName +
          ", productName=" + productName +
          ", quantity=" + quantity +
          ", price=" + price +
          ", orderDate=" + orderDate + ")";
  }
}
```

Producer.java

```
package com.example.orderprocessingsystem.service;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;
import com.example.orderprocessingsystem.entity.Order;
@Service
public class Producer {
	private static final String TOPIC = "order-topic";

	@Autowired
	private KafkaTemplate<String, Order> kafkaTemplate;

	public void sendOrder(Order order) {
		kafkaTemplate.send(TOPIC, order);
	}
}
```

Consumer.java

```
package com.example.orderprocessingsystem.service;
import org.springframework.kafka.annotation.KafkaListener;
import org.springframework.stereotype.Service;
import com.example.orderprocessingsystem.entity.Order;
@Service
public class Consumer {
	@KafkaListener(topics = "order-topic", groupId ="order-group", containerFactory =
"kafkaListenerContainerFactory")
	public void consumer(Order order) {
		System.out.println("Order received successfully: " +order);
	}
}
```

OrderController.java

```
package com.example.orderprocessingsystem.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.example.orderprocessingsystem.entity.Order;
import com.example.orderprocessingsystem.service.Producer;
@RestController
@RequestMapping("/api/orders")
public class OrderController {

	@Autowired
	private Producer producer;
```

```
    @PostMapping
    public String placeOrder(@RequestBody Order order) {
        producer.sendOrder(order);
        return "Order placed successfully";
    }
}
```

Output in the Console:

Received Order: Order(orderId=ORD101, customerName=John Doe, productName=Laptop, quantity=2, price=55000.0, orderDate=Fri Aug 08 05:30:00 IST 2025)