order-producer/pom.xml

```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-producer</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>
<name>order-producer</name>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.2.0</version>
<relativePath/>
</parent>

<properties>
<java.version>17</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.kafka</groupId>
<artifactId>spring-kafka</artifactId>
</dependency>

<dependency>
<groupId>com.fasterxml.jackson.core</groupId>
<artifactId>jackson-databind</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
</dependency>
```

```
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

order-producer/src/main/java/com/example/orderproducer/OrderProducerAp

```
package com.example.orderproducer;

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

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

order-producer/src/main/java/com/example/orderproducer/model/Order.java

```
package com.example.orderproducer.model;

import java.time.LocalDate;

public class Order {
```

```
private String orderId;
private String customerName;
private String productName;
private int quantity;
private double price;
private LocalDate orderDate;

public Order() {}

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

// getters and setters

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 java.time.LocalDate getOrderDate() { return orderDate; }
public void setOrderDate(java.time.LocalDate orderDate) { this.orderDate =
orderDate; }

@Override
public String toString() {
```

```
return "Order(orderId=" + orderId + ", customerName=" + customerName +
", productName=" + productName + ", quantity=" + quantity +
", price=" + ((int)price) + ", orderDate=" + orderDate + ")";
}
}
```

order-producer/src/main/java/com/example/orderproducer/config/KafkaProdu

```
package com.example.orderproducer.config;

import com.example.orderproducer.model.Order;
import org.apache.kafka.clients.producer.ProducerConfig;
import org.apache.kafka.common.serialization.StringSerializer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.core.*;
import org.springframework.kafka.support.serializer.JsonSerializer;

import java.util.HashMap;
import java.util.Map;

@Configuration
public class KafkaProducerConfig {

@Bean
public ProducerFactory<String, Order> producerFactory() {
Map<String, Object> configProps = new HashMap<>();
configProps.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG,
"localhost:9092");
configProps.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
StringSerializer.class);
configProps.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
JsonSerializer.class);
return new DefaultKafkaProducerFactory<>(configProps);
}

@Bean
public KafkaTemplate<String, Order> kafkaTemplate() {
return new KafkaTemplate<>(producerFactory());
}
}
```

order-producer/src/main/java/com/example/orderproducer/service/OrderProdu

```
package com.example.orderproducer.service;
```

```
import com.example.orderproducer.model.Order;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

@Service
public class OrderProducerService {

private static final String TOPIC = "order-topic";

@Autowired
private KafkaTemplate<String, Order> kafkaTemplate;

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

order-producer/src/main/java/com/example/orderproducer/controller/OrderCo

```
package com.example.orderproducer.controller;

import com.example.orderproducer.model.Order;
import com.example.orderproducer.service.OrderProducerService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.web.bind.annotation.*;

import java.time.LocalDate;

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

@Autowired
private OrderProducerService producerService;

@PostMapping
public String createOrder(@RequestBody OrderRequest request) {
Order order = new Order(
request.getOrderId(),
request.getCustomerName(),
request.getProductName(),
request.getQuantity(),
request.getPrice(),
request.getOrderDate()
);
producerService.sendOrder(order);
return "Order sent to Kafka";
}

static class OrderRequest {
private String orderId;
private String customerName;
private String productName;
private int quantity;
private double price;
@DateTimeFormat(iso = DateTimeFormat.ISO.DATE)
private LocalDate orderDate;

// getters and setters
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 LocalDate getOrderDate() { return orderDate; }
public void setOrderDate(LocalDate orderDate) { this.orderDate = orderDate; }
}
}
```

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

```
server.port=8080
spring.kafka.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer

spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeriali
```

order-consumer/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-consumer</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>
<name>order-consumer</name>

<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.2.0</version>
<relativePath/>
</parent>

<properties>
<java.version>17</java.version>
</properties>

<dependencies>
<dependency>
<groupId>org.springframework.kafka</groupId>
<artifactId>spring-kafka</artifactId>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId>
</dependency>

<dependency>
<groupId>com.fasterxml.jackson.core</groupId>
<artifactId>jackson-databind</artifactId>
</dependency>
```

```
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

order-consumer/src/main/java/com/example/orderconsumer/OrderConsumerA

```
package com.example.orderconsumer;

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

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

order-consumer/src/main/java/com/example/orderconsumer/model/Order.java

```
package com.example.orderconsumer.model;

import java.time.LocalDate;
```

```
public class Order {
private String orderId;
private String customerName;
private String productName;
private int quantity;
private double price;
private LocalDate orderDate;

public Order() {}

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 java.time.LocalDate getOrderDate() { return orderDate; }
public void setOrderDate(java.time.LocalDate orderDate) { this.orderDate =
orderDate; }

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

order-consumer/src/main/java/com/example/orderconsumer/config/KafkaCons

```
package com.example.orderconsumer.config;

import com.example.orderconsumer.model.Order;
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.springframework.kafka.support.serializer.JsonDeserializer;

import java.util.HashMap;
import java.util.Map;

@Configuration
public class KafkaConsumerConfig {

@Bean
public DefaultKafkaConsumerFactory<String, Order> consumerFactory() {
JsonDeserializer<Order> deserializer = new JsonDeserializer<>(Order.class);
deserializer.addTrustedPackages("*");
Map<String, Object> props = new HashMap<>();
props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
props.put(ConsumerConfig.GROUP_ID_CONFIG, "order-group");
props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class);
props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
JsonDeserializer.class);
return new DefaultKafkaConsumerFactory<>(props, new StringDeserializer(),
deserializer);
}

@Bean
public ConcurrentKafkaListenerContainerFactory<String, Order>
kafkaListenerContainerFactory() {
ConcurrentKafkaListenerContainerFactory<String, Order> factory =
new ConcurrentKafkaListenerContainerFactory<>();
factory.setConsumerFactory(consumerFactory());
return factory;
}
}
```

order-consumer/src/main/java/com/example/orderconsumer/service/OrderListe

```
package com.example.orderconsumer.service;

import com.example.orderconsumer.model.Order;
import org.springframework.kafka.annotation.KafkaListener;
import org.springframework.stereotype.Service;

@Service
public class OrderListener {

@KafkaListener(topics = "order-topic", groupId = "order-group", containerFactory =
"kafkaListenerContainerFactory")
public void listen(Order order) {
System.out.println("Received Order: " + order.toString());
// Simulate order processing: confirm stock, charge payment, etc.
}
}
```

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

```
server.port=8090
spring.kafka.bootstrap-servers=localhost:9092
spring.kafka.consumer.group-id=order-group

spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializ

spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDes
```