Order.java

```
package com.example.kafkademo.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;


    // 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; }

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

OrderProducer.java

```
package com.example.kafkademo.service;

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

@Service
public class OrderProducer {
```

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

    @Autowired
    private KafkaTemplate<String, Order> kafkaTemplate;

    public void sendOrder(Order order) {
        kafkaTemplate.send(TOPIC, order);
        System.out.println("Order sent to Kafka: " + order.getOrderId());
    }
}
```

OrderConsumer.java

```
package com.example.kafkademo.service;

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

@Service
public class OrderConsumer {

    @KafkaListener(topics = "order-topic", groupId = "order-group", containerFactory =
"orderListenerFactory")
    public void consume(Order order) {
        System.out.println("Received Order: " + order);
    }
}
```

KafkaConfig.java

```
package com.example.kafkademo.config;


import com.example.kafkademo.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.annotation.EnableKafka;

import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;

import org.springframework.kafka.core.*;


import java.util.HashMap;

import java.util.Map;


@Configuration

@EnableKafka

public class KafkaConfig {


   @Bean

   public ProducerFactory<String, Order> producerFactory() {

      Map<String, Object> config = new HashMap<>();

      config.put(org.apache.kafka.clients.producer.ProducerConfig.BOOTSTRAP_SERVERS_CONFIG,
"localhost:9092");

      config.put(org.apache.kafka.clients.producer.ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
org.apache.kafka.common.serialization.StringSerializer.class);
```

```
        config.put(org.apache.kafka.clients.producer.ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
org.springframework.kafka.support.serializer.JsonSerializer.class);
        return new DefaultKafkaProducerFactory<>(config);
    }

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

    @Bean
    public ConsumerFactory<String, Order> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        config.put(ConsumerConfig.GROUP_ID_CONFIG, "order-group");
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
org.springframework.kafka.support.serializer.JsonDeserializer.class);
        config.put(org.springframework.kafka.support.serializer.JsonDeserializer.TRUSTED_PACKAGES, "*");
        return new DefaultKafkaConsumerFactory<>(config);
    }

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

```
        return factory;
    }
}
```

OrderController.java

```
package com.example.kafkademo.controller;

import com.example.kafkademo.model.Order;
import com.example.kafkademo.service.OrderProducer;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

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

    @Autowired
    private OrderProducer producer;

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

KafkaDemoApplication.java

```
package com.example.kafkademo;
```

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class KafkaDemoApplication {

   public static void main(String[] args) {

      SpringApplication.run(KafkaDemoApplication.class, args);

   }

}
```

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>3.5.4</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.example</groupId>
	<artifactId>kafkademo</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>kafkademo</name>
	<description>Demo project for Apache Kafka</description>
	<url/>
	<licenses>
		<license/>
	</licenses>
	<developers>
		<developer/>
	</developers>
	<scm>
		<connection/>
		<developerConnection/>
		<tag/>
		<url/>
	</scm>
	<properties>
		<java.version>21</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>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
		<dependency>
			<groupId>org.springframework.kafka</groupId>
			<artifactId>spring-kafka-test</artifactId>
			<scope>test</scope>
		</dependency>
		<dependency>
	  <groupId>com.fasterxml.jackson.core</groupId>
	  <artifactId>jackson-databind</artifactId>
	</dependency>
	  </dependencies>

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

</project>
```

Application.properties

spring.kafka.bootstrap-servers=localhost:9092

spring.kafka.consumer.group-id=order-group

spring.kafka.consumer.auto-offset-reset=earliest

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

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

spring.kafka.consumer.properties.spring.json.trusted.packages=*

spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer

spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSerializer

```
        ssl.truststore.password = null
        ssl.truststore.type = JKS
        transaction.timeout.ms = 60000
        transactional.id = null
        value.serializer = class org.springframework.kafka.support.serializer.JsonSerializer

2025-08-08T16:50:07.597+05:30  INFO 11552 --- [kafkademo] [nio-8080-exec-2] o.a.k.c.t.i.KafkaMetricsCollector
2025-08-08T16:50:07.619+05:30  INFO 11552 --- [kafkademo] [nio-8080-exec-2] o.a.k.clients.producer.KafkaProducer
2025-08-08T16:50:07.651+05:30  INFO 11552 --- [kafkademo] [nio-8080-exec-2] o.a.kafka.common.utils.AppInfoParser
2025-08-08T16:50:07.651+05:30  INFO 11552 --- [kafkademo] [nio-8080-exec-2] o.a.kafka.common.utils.AppInfoParser
2025-08-08T16:50:07.651+05:30  INFO 11552 --- [kafkademo] [nio-8080-exec-2] o.a.kafka.common.utils.AppInfoParser
2025-08-08T16:50:07.665+05:30  INFO 11552 --- [kafkademo] [demo-producer-1] org.apache.kafka.clients.Metadata
2025-08-08T16:50:07.672+05:30  INFO 11552 --- [kafkademo] [demo-producer-1] o.a.k.c.p.internals.TransactionManage
Order sent to Kafka: ORD102
Received Order: Order(orderId=ORD102, customerName=Alice Johnson, productName=Smartphone, quantity=1, price=30000
```