**Sending and Receiving Messages with Apache Kafka**

**package** com.ecommerce;

**import** java.util.HashMap;

**import** java.util.Map;

**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.DefaultKafkaProducerFactory;

**import** org.springframework.kafka.core.KafkaTemplate;

**import** org.springframework.kafka.core.ProducerFactory;

**@Configuration**

**public** **class** KafkaProducerConfig {

**@Bean**

**public** ProducerFactory<**String**, **String**> 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, StringSerializer.class);

**return** **new** DefaultKafkaProducerFactory<>(configProps);

}

**@Bean**

**public** KafkaTemplate<**String**, **String**> kafkaTemplate() {

**return** **new** KafkaTemplate<>(producerFactory());

}

}

**package** com.ecommerce;

**import** java.util.HashMap;

**import** java.util.Map;

**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.ConsumerFactory;

**import** org.springframework.kafka.core.DefaultKafkaConsumerFactory;

**@EnableKafka**

**@Configuration**

**public** **class** KafkaConsumerConfig {

**@Bean**

**public** ConsumerFactory<**String**, **String**> consumerFactory() {

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

props.put(ConsumerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:2181");

props.put(ConsumerConfig.GROUP\_ID\_CONFIG, "group-id");

props.put(ConsumerConfig.KEY\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class);

props.put(ConsumerConfig.VALUE\_DESERIALIZER\_CLASS\_CONFIG, StringDeserializer.class);

**return** **new** DefaultKafkaConsumerFactory<>(props);

}

**@Bean**

**public** ConcurrentKafkaListenerContainerFactory<**String**, **String**> kafkaListenerContainerFactory() {

ConcurrentKafkaListenerContainerFactory<**String**, **String**>

factory = **new** ConcurrentKafkaListenerContainerFactory<>();

factory.setConsumerFactory(consumerFactory());

**return** factory;

}

}

**package** com.commerce.controllers;

**import** java.util.Calendar;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.kafka.core.DefaultKafkaProducerFactory;

**import** org.springframework.kafka.core.KafkaTemplate;

**import** org.springframework.kafka.core.ProducerFactory;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**@Controller**

**public** **class** MainController {

**@Autowired**

**private** KafkaTemplate<**String**, **String**> kafkaTemplate;

**@RequestMapping(value = "/")**

**public** **String** index() {

**this**.sendMessage("This is a message sent at " + **Calendar**.getInstance().getTime());

**return** "Check Eclipse console for kafka output";

}

**private** void sendMessage(**String** msg) {

kafkaTemplate.send("ecommerce", msg);

}

}

**package** com.ecommerce;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.boot.ApplicationArguments;

**import** org.springframework.boot.ApplicationRunner;

**import** org.springframework.boot.SpringApplication;

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

**import** org.springframework.kafka.annotation.KafkaListener;

**import** org.springframework.kafka.core.KafkaTemplate;

**@SpringBootApplication**

**public** **class** SpringBootKafkaApplication {

**@Autowired**

**private** KafkaTemplate<**String**, **String**> kafkaTemplate;

**public** **static** void main(**String**[] args) {

SpringApplication.run(SpringBootKafkaApplication.class, args);

}

**@KafkaListener(topics = "ecommerce", groupId = "group-id")**

**public** void listen(**String** message) {

**System**.out.println("Received Message in group - group-id: " + message);

}

}