Sending and Receiving Messages with Apache Kafka

Step 1: Creating a Spring Boot Starter Project which is web enabled and has Apache Kafka dependencies

```
<?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
http://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>2.1.6.RELEASE</version>

                <relativePath /> <!-- lookup parent from
repository -->

        </parent>

        <groupId>com.ecommerce</groupId>

        <artifactId>SpringKafka</artifactId>

        <version>0.0.1-SNAPSHOT</version>

        <name>SpringKafka</name>

        <description>Demo project for Spring Boot</description>


        <properties>

                <java.version>1.8</java.version>

        </properties>
```

```
<dependencies>
        <dependency>
            <groupId>
org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>org.apache.kafka</groupId>
            <artifactId>kafka-streams</artifactId>
        </dependency>
        <dependency>
            <groupId>
org.springframework.kafka</groupId>
            <artifactId>spring-kafka</artifactId>
            <version>2.1.6.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-web</artifactId>
            <version>5.1.5.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>
org.springframework.kafka</groupId>
            <artifactId>spring-kafka</artifactId>
            <version>2.2.7.RELEASE</version>
```

```
            </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>
```

Step 2: Creating a KafkaProducerConfig class

```
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_CONFI
G, StringSerializer.class);

configProps.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CO
NFIG, StringSerializer.class);
    return new DefaultKafkaProducerFactory<>(configProps);
  }
  @Bean
  public KafkaTemplate<String, String> kafkaTemplate() {
    return new KafkaTemplate<>(producerFactory());
  }
}
```

Step 3: Creating a KafkaConsumerConfig class

```
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.ConcurrentKafkaListenerContainer
Factory;
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_CONF
IG, StringDeserializer.class);
    return new DefaultKafkaConsumerFactory<>(props);
  }
  @Bean
  public ConcurrentKafkaListenerContainerFactory<String, String>
kafkaListenerContainerFactory() {
    ConcurrentKafkaListenerContainerFactory<String, String>
    factory = new ConcurrentKafkaListenerContainerFactory<>();
    factory.setConsumerFactory(consumerFactory());
    return factory;
  }
```

```
}
```

Step 4: Creating MainController to send a Kafka message

```
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);
	  }

}
```

Step 5: Configuring SpringRestApplication to listen to Kafka messages

```
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);
	  }

}
```