Java Programming
Lab Kafka 4

1. Introduction

In this lab, you will add a Java Consumer to read from the lab3 topic you used in the last lab while using the Producer you created in the last lab to write to the topic.

2. Setup

Create a new Maven project called Consumer and follow the steps from Lab 2 to set up the POM.xml life and create the project. In the lab solution the cor-ordinates used were “kafka” and “Consumer”

Do not delete or modify the project from the last lab, you will use it in this lab to test your code.

3. Create the code

In a package called **consume**, create a new class with a main() method called Consumer. The code that we will be adding looks a lot like the code we wrote for the producers.

First set up the configuration for the consumer.

```
public class Consumer {

    public static void main(String[] args) {
        Properties prop = new Properties();
        prop.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        prop.put(ConsumerConfig.GROUP_ID_CONFIG, "Group 1");
        prop.put(ConsumerConfig.CLIENT_ID_CONFIG, "Consumer");
        prop.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,StringDeserializer.class.getName());
        prop.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,StringDeserializer.class.getName());

        KafkaConsumer<String,String> consumer = new KafkaConsumer<>(prop);
        ConsumerRecords<String,String> records ;
```

In the next part, we subscribe to a list of topics, in this case the list only has one element, our lab3 topic.

In the next part, we are polling the topic in a basically do-forever sort of loop. Each time we poll the topic, we print out the number of messages received and then print them out. The loop will end when we get an exception thrown in this case, like when the server is stopped and the topic in not available.

```
try {
    while(true) {
        records = consumer.poll(Duration.ofMillis(1000));
        System.out.println("Got " + records.count() + " messages");
        for (ConsumerRecord<String,String> record : records ) {
            System.out.println("Message: " + record);
        }

    }
}
finally {
    consumer.close();
}
```

In a more production sort of application, the messages received would be dispatched somewhere for processing by another thread.

4. Run and test the code

First, if it's not running, start zookeeper and the kafka server. At this point it is assumed that the **lab3** topic created in Lab 3 is still there (you should know how to check on it at this point). You don't need a console producer or consumer, the two Java applications should be enough.

Open up the producer you created in the last lab so that you can run it when necessary.

Start up the consumer you just wrote as a Java application. Notice that it keeps returning 0 records since nothing has been written to the topic since it started.

Without stopping the consumer, run the producer from lab 3.

You should see output that looks like the image on the next page. Hit the red square button to stop the consumer from running.

```
Got 0 messages
Got 0 messages
Got 0 messages
Got 0 messages
Got 0 messages
Got 0 messages
Got 0 messages
Got 0 messages
Got 10 messages
Message: ConsumerRecord(topic = lab3, partition = 5, leaderEpoch = 0, offset = 3, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 4, leaderEpoch = 0, offset = 3, CreateTime = 1675033220924, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 7, leaderEpoch = 0, offset = 6, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 7, leaderEpoch = 0, offset = 7, CreateTime = 1675033220933, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 6, leaderEpoch = 0, offset = 9, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 6, leaderEpoch = 0, offset = 10, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11,
Message: ConsumerRecord(topic = lab3, partition = 6, leaderEpoch = 0, offset = 11, CreateTime = 1675033220933, serialized key size = 7, serialized value size = 11,
Message: ConsumerRecord(topic = lab3, partition = 0, leaderEpoch = 0, offset = 3, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 8, leaderEpoch = 0, offset = 6, CreateTime = 1675033220932, serialized key size = 7, serialized value size = 11, h
Message: ConsumerRecord(topic = lab3, partition = 8, leaderEpoch = 0, offset = 7, CreateTime = 1675033220933, serialized key size = 8, serialized value size = 12, h
Got 0 messages
Got 0 messages
Got 0 messages
```

With the actual messages (on the far right)

```
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 5: , value = Message - 5)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 1: , value = Message - 1)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 6: , value = Message - 6)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 9: , value = Message - 9)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 3: , value = Message - 3)
serialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 7: , value = Message - 7)
serialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 8: , value = Message - 8)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 2: , value = Message - 2)
erialized value size = 11, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 4: , value = Message - 4)
erialized value size = 12, headers = RecordHeaders(headers = [], isReadOnly = false), key = Key 10: , value = Message - 10)
```

Clean up the lab by shutting down Kafka

End of Lab