

Java Programming

Lab Kafka 3

1. Introduction

In this lab, you will modify the producer from Lab two to sent records with different transmission properties and topic that accepts key values.

Our new topic will be lab3 and it will have 10 partitions. Now remember that we don't have to have as many partitions as we do key values, the keys are like hash values so that all messages with the same key will always be in the same partition, but a partition may have more than one key value among its messages.

2. Setup

Set up the project for this lab, the same way you did for the last one. The changes we are making in configuration is that this will be in package "prod2" with the main class "Producer2."

Start up Kafka like you did before and create a new topic called "lab3" with 10 partitions like so:

```
C:\kafka>.\bin\windows\kafka-topics.bat --create --topic lab3 --partitions 10 --bootstrap-server localhost:9092
Created topic lab3.

C:\kafka>.\bin\windows\kafka-topics.bat --describe --topic lab3 --bootstrap-server localhost:9092
Topic: lab3      TopicId: hOMrRw8gRkiiGNJBzeGB3A PartitionCount: 10      ReplicationFactor: 1      Configs:
  Topic: lab3    Partition: 0    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 1    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 2    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 3    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 4    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 5    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 6    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 7    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 8    Leader: 0      Replicas: 0      Isr: 0
  Topic: lab3    Partition: 9    Leader: 0      Replicas: 0      Isr: 0
```

And create a console consumer to read off of the topic but we will also print out the key associated with each message by adding the "--property print.key=true" to the command like so:

```
C:\kafka>.\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic lab3 --property print.key=true
```

Part 2. Modify the code

In the previous lab, the producer sent messages one at a time all into the same partition. In the properties, we now batch up our messages before we send them and want the confirmation that the messages are written to their partition and replica. We also set the size of the output buffer.

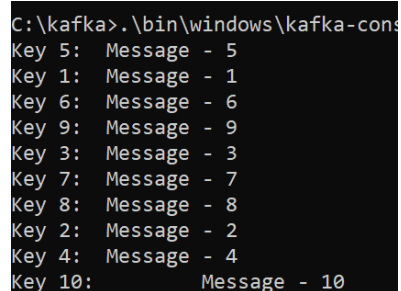
```
public static void main(String[] args) {
    Properties prop = new Properties();
    prop.setProperty("bootstrap.servers", "localhost:9092");
    prop.setProperty("key.serializer", StringSerializer.class.getName());
    prop.setProperty("value.serializer", StringSerializer.class.getName());
    prop.put("client.id", "Producer2");
    prop.put("acks", "all");
    prop.put("retries", 0);
    prop.put("batch.size", 16384);
    prop.put("buffer.memory", 33554432);
```

Now we use a different form of producer record with with the key and value being strings. Everything else is the same as the previous lab.

```
final KafkaProducer<String,String> p = new KafkaProducer<String,String>(prop);
ProducerRecord<String,String> record;

for (int i = 1 ; i <=10; i++) {
    String key = "Key " + i + ": ";
    record = new ProducerRecord<>("lab3",key,"Message - "+i);
    // Send data
    p.send(record);
    System.out.println("Sent record " + i);
}
// flush and close the producer
p.flush();
p.close();
```

Run the code then check the output on the console consumer



```
C:\kafka>. \bin\windows\kafka-cons
Key 5: Message - 5
Key 1: Message - 1
Key 6: Message - 6
Key 9: Message - 9
Key 3: Message - 3
Key 7: Message - 7
Key 8: Message - 8
Key 2: Message - 2
Key 4: Message - 4
Key 10: Message - 10
```

End of Lab