

Session 23: Assignment 1 | More Kafka Assignment

Task 1:

Create a java program `MyKafkaProducer.java` that takes a file name and delimiter as input arguments.

It should read the content of file line by line.

Fields in the file are in following order

1. Kafka Topic Name
2. Key
3. value


For every line, insert the key and value to the respective Kafka broker in a fire and forget mode.

After record is sent, it should print appropriate message on screen.

Pass `dataset_producer.txt` as the input file and `-as` delimiter.

LINK: https://drive.google.com/file/d/0B_Qjau8wv1KoSnR5eHpKOF9rTFU/view?usp=sharing

Please note the link does not work. I will create my own file called `dataset_producer.txt` instead.

 https://drive.google.com/file/d/0B_Qjau8wv1KoSnR5eHpKOF9rTFU/view?usp=sharing



Sorry, the file you have requested does not exist.

Make sure that you have the correct URL and the file exists.

Get stuff done with Google Drive

Apps in Google Drive make it easy to create, store and share online documents, spreadsheets, presentations and more.

Learn more at drive.google.com/start/apps.

File: dataset_producer

```
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Session23]$ cat dataset_producer.txt
Captain | Picard
Captain | Kirk
Captain | Janeway
Captain | Archer
Captain | McClain
FirstOfficer | Tapol
FirstOfficer | Chakotay
FirstOfficer | Riker
FirstOfficer | Spock
ChiefEngineer | Laforge
ChiefEngineer | Tucker
ChiefEngineer | Torres
StarShip | Enterprise NCC-1701
StarShip | Enterprise NCC-1701-B
StarShip | Enterprise NX-01
StarShip | Enterprise NCC-1701-A
StarShip | Enterprise NCC-1701-C
StarShip | Enterprise NCC-1701-D
[acadgild@localhost Session23]$
```

Kafka Command:

```

You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Session23]$ java -cp $KAFKA_HOME/libs/*:./session23Assgn1.jar
MyKafkaProducer dataset_producer.txt \
Index 0: Captain
Index 1: Picard
Index 0: Captain
Index 1: Kirk
Index 0: Captain
Index 1: Janeway
Index 0: Captain
Index 1: Archer
Index 0: Captain
Index 1: McClain
Index 0: FirstOfficer
Index 1: Tapol
Index 0: FirstOfficer
Index 1: Chakotay
Index 0: FirstOfficer
Index 1: Riker
Index 0: FirstOfficer
Index 1: Spock
Index 0: ChiefEngineer
Index 1: Laforge
Index 0: ChiefEngineer
Index 1: Tucker
Index 0: ChiefEngineer
Index 1: Torres
Index 0: StarShip
Index 1: Enterprise NCC-1701
Index 0: StarShip
Index 1: Enterprise NCC-1701-B
Index 0: StarShip
Index 1: Enterprise NX-01
Index 0: StarShip
Index 1: Enterprise NCC-1701-A
Index 0: StarShip
Index 1: Enterprise NCC-1701-C
Index 0: StarShip
Index 1: Enterprise NCC-1701-D
Index 0:

```

Output

```

[2019-02-18 08:02:00,805] WARN ClientSession timed out, have not heard from ser
ver in 5016ms for sessionid 0x1685cee008f0012 (org.apache.zookeeper.ClientCnxn)
Captain      Picard
Captain      Picard
Captain      Kirk
Captain      Janeway
Captain      Archer
Captain      McClain
ChiefEngineer Laforge
ChiefEngineer Tucker
ChiefEngineer Torres
FirstOfficer  Tapol
FirstOfficer  Chakotay
FirstOfficer  Riker
FirstOfficer  Spock
StarShip      Enterprise NCC-1701
StarShip      Enterprise NCC-1701-B
StarShip      Enterprise NX-01
StarShip      Enterprise NCC-1701-A
StarShip      Enterprise NCC-1701-C
StarShip      Enterprise NCC-1701-D
^CProcessed a total of 56 messages
You have new mail in /var/spool/mail/acadgild

```

Jar file:

```
[acadgild@localhost Session23]$ ls
dataset_producer.txt  PartitionedProducer.java  session23Assgn1.jar
kafka_command.txt    partproducer.jar
Maven                 RangePartitioner.java
[acadgild@localhost Session23]$ |
```

Java Program: KafkaProducer.java

```
import java.util.Properties;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.File;

public class MyKafkaProducer {

    public static void main(String[] args) {
        File inFile = null;
        String delimiter = ","; //default to comma

        if (args.length != 2) {
            System.out.println("Require <<filename>> <<delimiter>>");
            System.exit(-1);
        }
        else {
            inFile = new File(args[0]);
            if (args[1] == "\\") {
                delimiter = "\\" + args[1];
            } else {
                delimiter = args[1];
            }
        }

        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new FileReader(inFile));
            Properties props = new Properties();
            props.put("bootstrap.servers", "localhost:9092");
            props.put("acks", "0");
            props.put("retries", 0);
            props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
            props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
            //props.put("partitioner.class", "RangePartitioner");
            props.put("MyKafkaProcessor.class", "MyKafkaProcessor");

            String topic = "PartitionedProducer";
            KafkaProducer<String, String> producer = new KafkaProducer<>(props);
            ProducerRecord<String, String> producerRecord = null;
            String line;
            String[] keyvalue;

            while ((line = reader.readLine()) != null) {
                keyvalue = null;
                keyvalue = line.split(delimiter);
                System.out.println("Index 0: " + keyvalue[0].toString());
                System.out.println("Index 1: " + keyvalue[1].toString());
                producerRecord = new ProducerRecord<String, String>(topic, new String(keyvalue[0]), keyvalue[1]);
                producer.send(producerRecord);
            }

            producer.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Program : MyKafkaProcessor

```
acaogio@iocainost:~/eclipse-workspace/session23/src
import java.util.List;
import java.util.Map;

import org.apache.kafka.clients.producer.Partitioner;
import org.apache.kafka.common.Cluster;
import org.apache.kafka.common.PartitionInfo;

public class MyKafkaProcessor implements Partitioner {

    public void configure(Map<String, ?> configs) {}

    public int partition(String topic, Object key, byte[] keyBytes,
        Object value, byte[] valueBytes, Cluster cluster) {
        //List<PartitionInfo> partitions = cluster.partitionsForTopic(topic);

        // Getting the number of partitions for the topic
        //int numPartitions = partitions.size();
        //if ((Integer) key < 5) {
        //    System.out.println("The key: " + key + " will go to partition: " + (numPartitions - 1));
        //    return numPartitions - 1;
        //}
        System.out.println("key: " + key + " value: " + value);
        //return numPartitions;
        return 0;
    }

    public void close() {}
}

~
~
~
~
```

Task 2:

Modify the previous program `MyKafkaProducer.java` and create a new Java program `KafkaProducerWithAck.java`

This should perform the same task as of `KafkaProducer.java` with some modification.

When passing any data to a topic, it should wait for acknowledgement.

After acknowledgement is received from the broker, it should print the key and value which has been written to a specified topic.

The application should attempt for 3 retries before giving any exception.

Pass `dataset_producer.txt` as the input file and `-as` delimiter.

Source: KafkaProducerWithAck.java

```
import java.util.Properties;
import java.util.concurrent.ExecutionException;

import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;
import org.apache.kafka.clients.producer.RecordMetadata;
import org.apache.kafka.clients.producer.Callback;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.File;

public class KafkaProducerWithAck {

    public static void main(String[] args) {
        File inFile = null;
        String delimiter = ","; //default to comma

        if (args.length != 2) {
            System.out.println("Require <<filename>> <<delimiter>>");
            System.exit(-1);
        }
        else {
            inFile = new File(args[0]);
            if (args[1] == "\\") {
                delimiter = "\\" + args[1];
            } else {
                delimiter = args[1];
            }
        }
        BufferedReader reader = null;
        try {
            reader = new BufferedReader(new FileReader(inFile));
            Properties props = new Properties();
            props.put("bootstrap.servers", "localhost:9092");
            props.put("acks", "all");
            props.put("retries", 3);
            props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
            props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
            //props.put("partitioner.class", "RangePartitioner");
            props.put("MyKafkaProcessor.class", "MyKafkaProcessor");

            String topic = "PartitionedProducer2";
            KafkaProducer<String, String> producer = new KafkaProducer<>(props);
            ProducerRecord<String, String> producerRecord = null;
            String line;
            String[] keyvalue;

            while ((line = reader.readLine()) != null) {
                keyvalue = null;
                keyvalue = line.split("\\|");
                System.out.println("Index 0: " + keyvalue[0].toString());
                System.out.println("Index 1: " + keyvalue[1].toString());
                producerRecord = new ProducerRecord<String, String>(topic, new String(keyvalue[0]), keyvalue[1]);

                RecordMetadata metadata = producer.send(producerRecord).get();
                System.out.printf("sent record(key=%s value=%s) \n", producerRecord.key(), producerRecord.value());
            }

            producer.close();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (ExecutionException ee) {
            ee.printStackTrace();
        } catch (InterruptedException ie) {
            ie.printStackTrace();
        }
    }
}

~
~
~
"KafkaProducerWithAck.java" 79L, 2369C
```

Program: KafkaProcessor.java

```

import java.util.List;
import java.util.Map;

import org.apache.kafka.clients.producer.Partitioner;
import org.apache.kafka.common.Cluster;
import org.apache.kafka.common.PartitionInfo;

import org.apache.kafka.clients.producer.Partitioner;

public class KafkaProcessor implements Partitioner {

    public void configure(Map<String, ?> configs) {}

    public int partition(String topic, Object key, byte[] keyBytes,
        Object value, byte[] valueBytes, Cluster cluster) {
        //List<PartitionInfo> partitions = cluster.partitionsForTopic(topic);

        // Getting the number of partitions for the topic
        //int numPartitions = partitions.size();
        //if (((Integer) key > 1) {
        //    System.out.println("The key: " + key + " will go to partitions: " + (numPartitions - 1));
        //    return numPartitions - 1;
        //}

        System.out.println("key: " + key + " value: " + value);
        //return numPartitions;
        return 0;
    }

    public void close() {}
}

```

"KafkaProcessor.java" 31L, 1146C

Data

```
[acadgild@localhost Session23]$ cat dataset_producer.txt
Captain | Picard
Captain | Kirk
Captain | Janeway
Captain | Archer
Captain | McClain
FirstOfficer | Tapol
FirstOfficer | Chakotay
FirstOfficer | Riker
FirstOfficer | Spock
ChiefEngineer | Laforge
ChiefEngineer | Tucker
ChiefEngineer | Torres
StarShip | Enterprise NCC-1701
StarShip | Enterprise NCC-1701-B
StarShip | Enterprise NX-01
StarShip | Enterprise NCC-1701-A
StarShip | Enterprise NCC-1701-C
StarShip | Enterprise NCC-1701-D
```

Receiver Topic: PartitionedProducer2

```
bin config ns_err_producer.log /usr/share/licenses/notice/notice.html
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-topics.sh --list --zookeep
er localhost:2181
PartitionedProducer - marked for deletion
PartitionedProducer1
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-topics.sh --create --topic
PartitionedProducer2 --zookeeper localhost:2181 --partitions 2 --replication-fa
ctor 1
Created topic "PartitionedProducer2".
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-console-consumer.sh --topi
c PartitionedProducer2 --from-beginning --zookeeper localhost:2181 --property pri
nt.key=true --property key.deserializer=org.apache.kafka.common.serialization.St
ringDeserializer
from-beginning is not a recognized option
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost kafka_2.12-0.10.1.1]$ ./bin/kafka-console-consumer.sh --topi
c PartitionedProducer2 --from-beginning --zookeeper localhost:2181 --property pr
int.key=true --property key.deserializer=org.apache.kafka.common.serialization.S
tringDeserializer
Using the ConsoleConsumer with old consumer is deprecated and will be removed in
a future major release. Consider using the new consumer by passing [bootstrap-s
erver] instead of [zookeeper].
Captain      Picard
Captain      Kirk
Captain      Janeway
Captain      Archer
Captain      McClain
FirstOfficer  Tapol
FirstOfficer  Chakotay
FirstOfficer  Riker
ChiefEngineer Laforge
ChiefEngineer Tucker
FirstOfficer  Spock
ChiefEngineer Torres
StarShip     Enterprise NCC-1701
StarShip     Enterprise NCC-1701-B
StarShip     Enterprise NX-01
StarShip     Enterprise NCC-1701-A
StarShip     Enterprise NCC-1701-C
StarShip     Enterprise NCC-1701-D
```

Output from running jar

```
[acadgild@localhost Session23]$ java -cp $KAFKA_HOME/libs/*:./session23Assgn1Ack
.jar KafkaProducerWithAck dataset_producer.txt \
Index 0: Captain
Index 1: Picard
sent record(key=Captain value= Picard )
Index 0: Captain
Index 1: Kirk
sent record(key=Captain value= Kirk )
Index 0: Captain
Index 1: Janeway
sent record(key=Captain value= Janeway)
Index 0: Captain
Index 1: Archer
sent record(key=Captain value= Archer )
Index 0: Captain
Index 1: McClain
sent record(key=Captain value= McClain )
Index 0: FirstOfficer
Index 1: Tapol
sent record(key=FirstOfficer value= Tapol)
Index 0: FirstOfficer
Index 1: Chakotay
sent record(key=FirstOfficer value= Chakotay)
Index 0: FirstOfficer
Index 1: Riker
sent record(key=FirstOfficer value= Riker)
Index 0: FirstOfficer
Index 1: Spock
sent record(key=FirstOfficer value= Spock )
Index 0: ChiefEngineer
Index 1: Laforge
sent record(key=ChiefEngineer value= Laforge)
Index 0: ChiefEngineer
Index 1: Tucker
sent record(key=ChiefEngineer value= Tucker )
Index 0: ChiefEngineer
Index 1: Torres
sent record(key=ChiefEngineer value= Torres)
Index 0: StarShip
Index 1: Enterprise NCC-1701
sent record(key=StarShip value= Enterprise NCC-1701)
Index 0: StarShip
Index 1: Enterprise NCC-1701-B
sent record(key=StarShip value= Enterprise NCC-1701-B)
Index 0: StarShip
Index 1: Enterprise NX-01
sent record(key=StarShip value= Enterprise NX-01)
Index 0: StarShip
Index 1: Enterprise NCC-1701-A
sent record(key=StarShip value= Enterprise NCC-1701-A)
Index 0: StarShip
Index 1: Enterprise NCC-1701-C
sent record(key=StarShip value= Enterprise NCC-1701-C)
Index 0: StarShip
Index 1: Enterprise NCC-1701-D
sent record(key=StarShip value= Enterprise NCC-1701-D )
You have new mail in /var/spool/mail/acadgild
[acadgild@localhost Session23]$
```

