

Assignment Day 23

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/oB_Qjau8wv1KoSnR5eHpKOF9rTFU/view?usp=sharing

Note: Associated text file is properly documented & submitted along with this file. Kindly refer the text file for a detailed description of each instruction used within the complete process.

Ans:

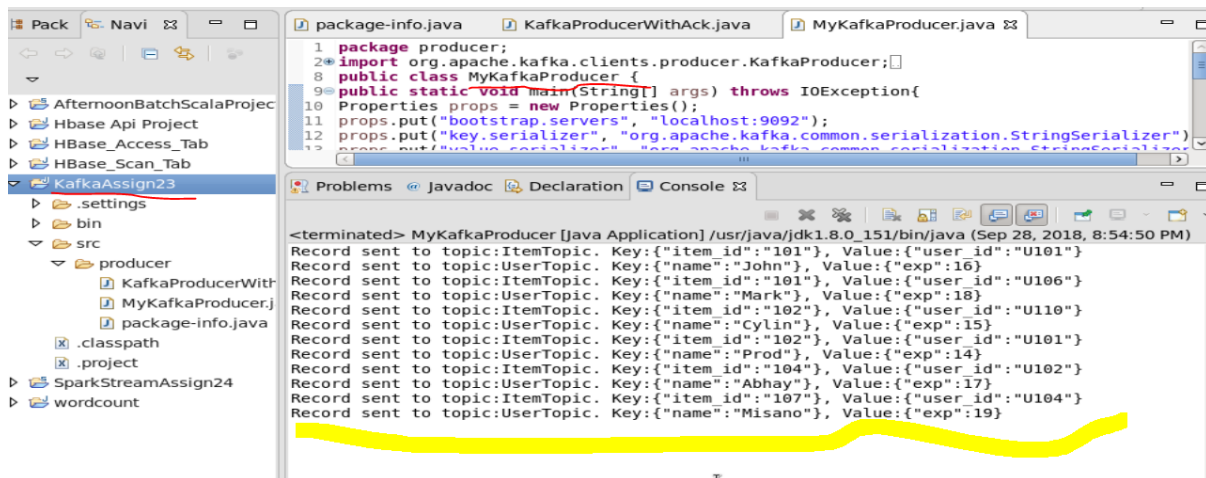
##Zookeeper Start Command

```
$KAFKA_HOME/bin/zookeeper-server-start.sh \  
$KAFKA_HOME/config/zookeeper.properties
```

Starting broker

```
$KAFKA_HOME/bin/kafka-server-start.sh \  
$KAFKA_HOME/config/server.properties
```

ScreenShot:



The screenshot shows an IDE with a project named 'KafkaAssign23'. The project structure includes a 'src' directory with files 'KafkaProducerWithAck.java', 'MyKafkaProducer.java', and 'package-info.java'. The 'Console' tab is active, displaying the output of the 'MyKafkaProducer' application. The output shows records being sent to 'ItemTopic' and 'UserTopic' with keys and values. A yellow highlight is under the last line of the console output: 'Record sent to topic:UserTopic. Key:{"name":"Misano"}, Value:{"exp":19}'.

```
1 package producer;
2 import org.apache.kafka.clients.producer.KafkaProducer;
3 public class MyKafkaProducer {
4     public static void main(String[] args) throws IOException{
5         Properties props = new Properties();
6         props.put("bootstrap.servers", "localhost:9092");
7         props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
8         props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
9         KafkaProducer<String, String> producer = new KafkaProducer<>(props);
10        producer.send(new ProducerRecord<>("ItemTopic", "101", "John", "U101"));
11        producer.send(new ProducerRecord<>("UserTopic", "101", "John", "U101"));
12        producer.send(new ProducerRecord<>("ItemTopic", "101", "Mark", "U106"));
13        producer.send(new ProducerRecord<>("UserTopic", "101", "Mark", "U106"));
14        producer.send(new ProducerRecord<>("ItemTopic", "102", "Cylin", "U110"));
15        producer.send(new ProducerRecord<>("UserTopic", "102", "Cylin", "U110"));
16        producer.send(new ProducerRecord<>("ItemTopic", "102", "Prod", "U101"));
17        producer.send(new ProducerRecord<>("UserTopic", "102", "Prod", "U101"));
18        producer.send(new ProducerRecord<>("ItemTopic", "104", "Abhay", "U102"));
19        producer.send(new ProducerRecord<>("UserTopic", "104", "Abhay", "U102"));
20        producer.send(new ProducerRecord<>("ItemTopic", "107", "Misano", "U104"));
21        producer.send(new ProducerRecord<>("UserTopic", "107", "Misano", "U104"));
22    }
23 }
```

```
<terminated> MyKafkaProducer [Java Application] /usr/java/jdk1.8.0_151/bin/java (Sep 28, 2018, 8:54:50 PM)
Record sent to topic:ItemTopic. Key:{"item_id":"101"}, Value:{"user_id":"U101"}
Record sent to topic:UserTopic. Key:{"name":"John"}, Value:{"exp":16}
Record sent to topic:ItemTopic. Key:{"item_id":"101"}, Value:{"user_id":"U106"}
Record sent to topic:UserTopic. Key:{"name":"Mark"}, Value:{"exp":18}
Record sent to topic:ItemTopic. Key:{"item_id":"102"}, Value:{"user_id":"U110"}
Record sent to topic:UserTopic. Key:{"name":"Cylin"}, Value:{"exp":15}
Record sent to topic:ItemTopic. Key:{"item_id":"102"}, Value:{"user_id":"U101"}
Record sent to topic:UserTopic. Key:{"name":"Prod"}, Value:{"exp":14}
Record sent to topic:ItemTopic. Key:{"item_id":"104"}, Value:{"user_id":"U102"}
Record sent to topic:UserTopic. Key:{"name":"Abhay"}, Value:{"exp":17}
Record sent to topic:ItemTopic. Key:{"item_id":"107"}, Value:{"user_id":"U104"}
Record sent to topic:UserTopic. Key:{"name":"Misano"}, Value:{"exp":19}
```

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.

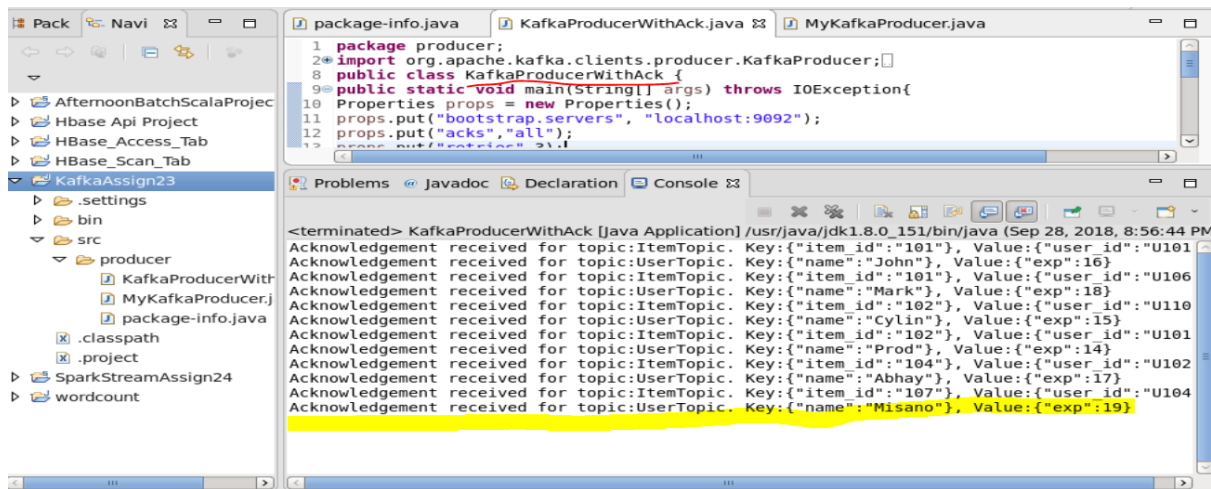
The application should attempt for 3 retries before giving any exception.

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

Ans:

Note: Associated text file is properly documented & submitted along with this file. Kindly refer the text file for a detailed description of each instruction used within the complete process.

ScreenShot:



End
