Apache Kafka - Simple Producer Example

### **The Producer Class**

The producer class provides send method to **send** messages to either single or multiple topics using the following signatures.

public void send(KeyedMessaget<k,v> message)

- sends the data to a single topic,par-titioned by key using either sync or async producer.

public void send(List<KeyedMessage<k,v>>messages)

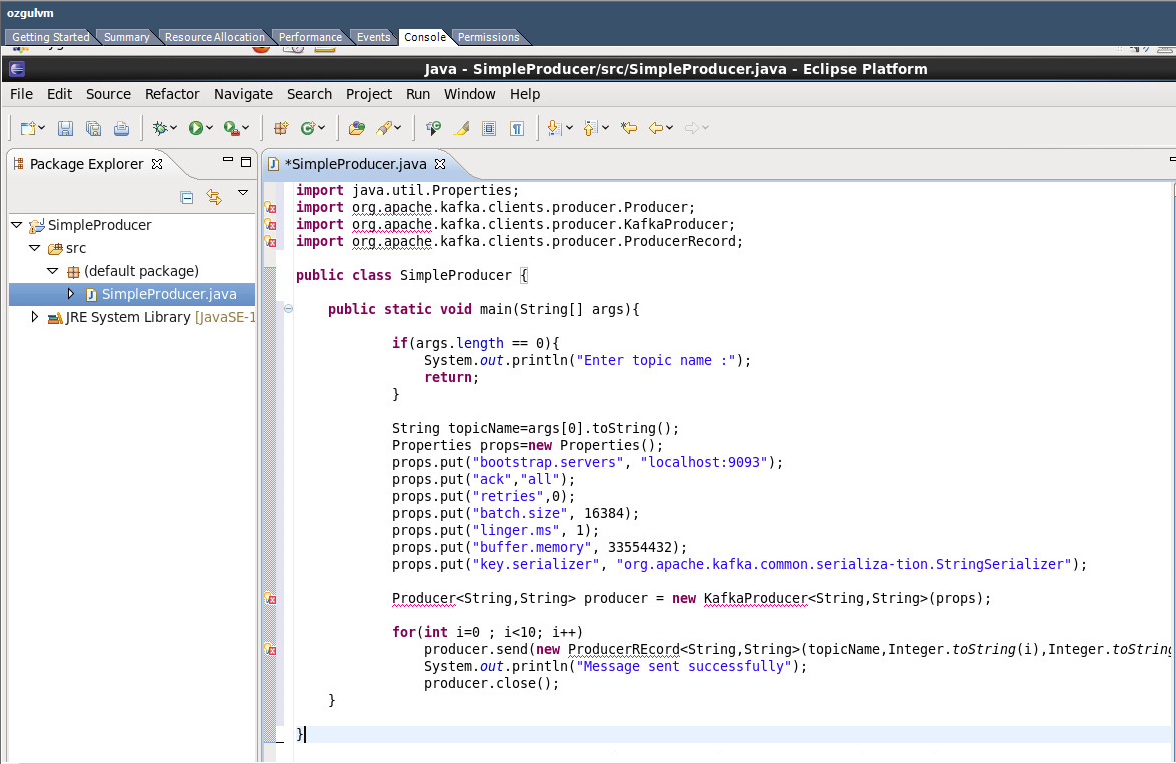
- sends data to multiple topics.

Properties prop = new Properties();

prop.put(producer.type,”async”)

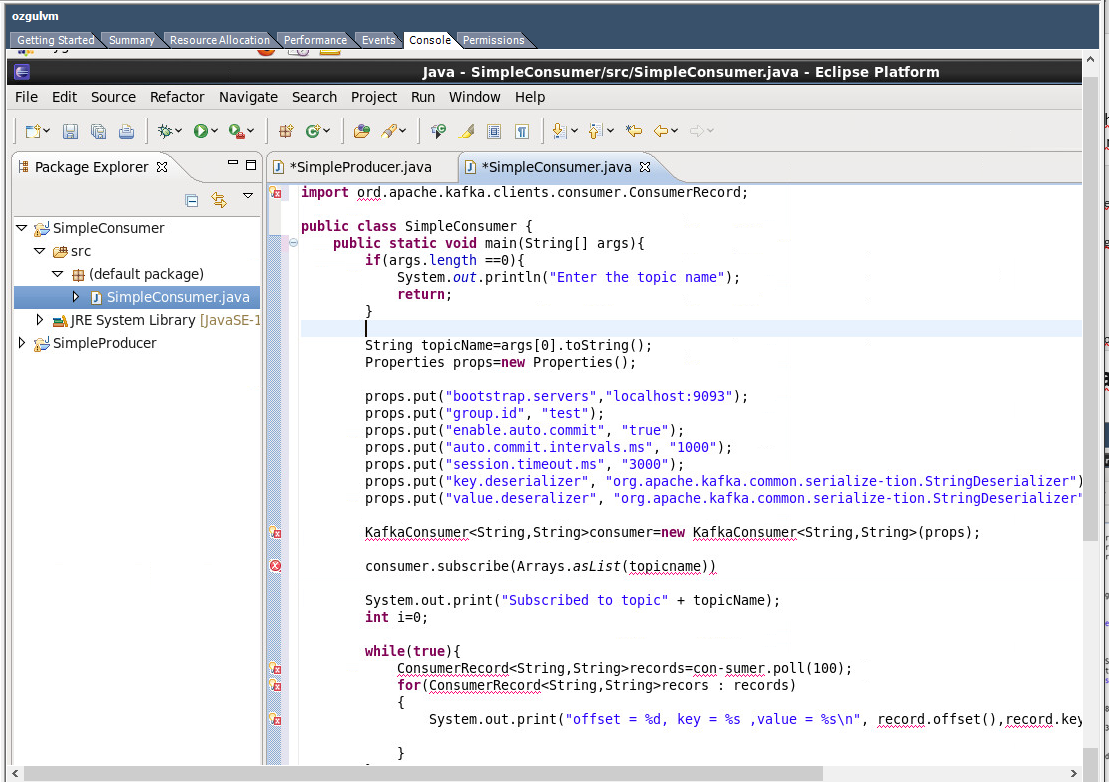
ProducerConfig config = new ProducerConfig(prop);

## **SimpleProducer application**



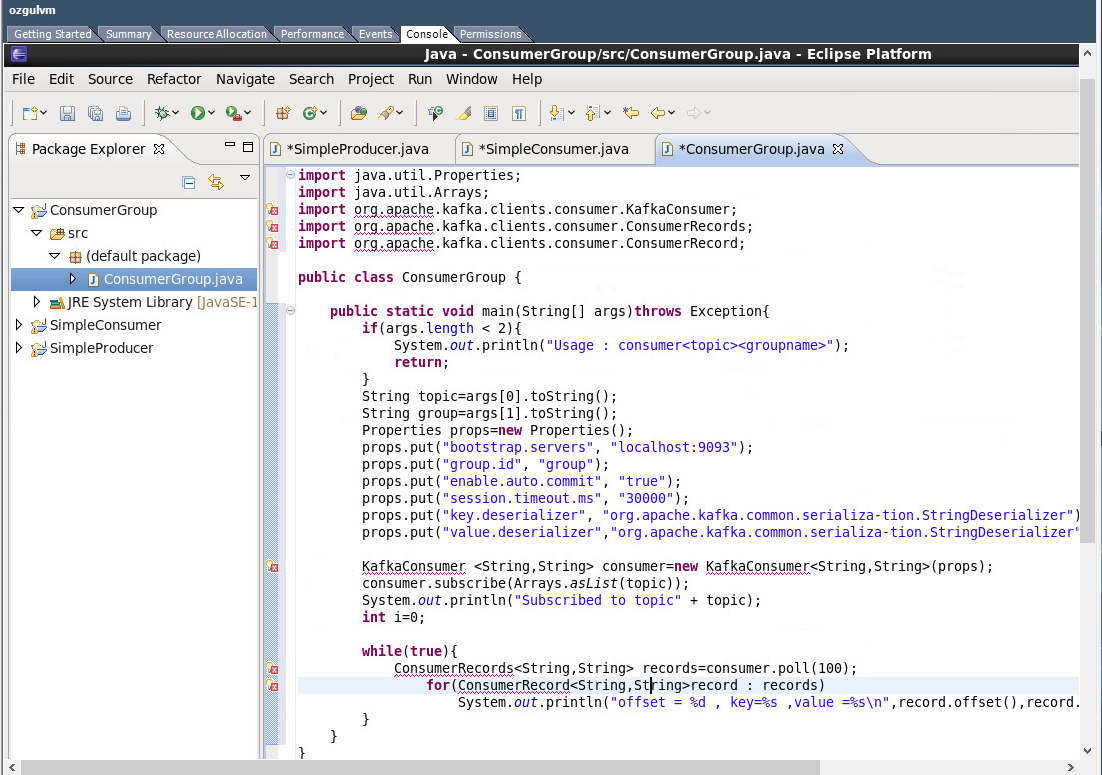
## **SimpleConsumer Application**

The producer application steps remain the same here. First, start your ZooKeeper and Kafka broker. Then create a SimpleConsumer application with the java class named SimpleCon-sumer.java and type the following code.



## **Consumer Group Example**

* Consumers can join a group by using the samegroup.id.
* The maximum parallelism of a group is that the number of consumers in the group ← no of partitions.
* Kafka assigns the partitions of a topic to the consumer in a group, so that each partition is consumed by exactly one consumer in the group.
* Kafka guarantees that a message is only ever read by a single consumer in the group.
* Consumers can see the message in the order they were stored in the log.



Bugün de dün öğrendiklerimi pratiğe döktüm.Bu kodları yazıp çalıştırarak Kafka’nın en önemli işlevlerinin nasıl çalıştığını öğrendim.Bundan sonra daha karmaşık yapılar üzerinde kodlama yapacağım.

Today I have practiced what I learned yesterday. By writing and running these codes, I learned how the most important functions of Kafka work. After that I will code on more complicated structures.