

# Custom Partitioner

## CountryPartitioner.java

## implements Partitioner

```
vim /home/hadoop/Admatic/Kafka/java/Kafka-Producer/src/main/java/com/admatic/CountryPartitioner.java
```

```
package com.admatic;

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

import java.util.HashMap;
import java.util.Map;

public class CountryPartitioner implements Partitioner {
    private static Map<String, Integer> countryToPartitionMap;

    // This method will gets called at the start, you should use it to do one time startup activity
    public void configure(Map<String, ?> configs) {
        countryToPartitionMap = new HashMap<>();
        for (Map.Entry<String, ?> entry : configs.entrySet()) {
            if (entry.getKey().startsWith("partitions."))
            {
                String key = entry.getKey();
                String value = (String) entry.getValue();

                int partitionId = Integer.parseInt(key.substring(11));
                countryToPartitionMap.put(value, partitionId);
            }
        }
    }

    //This method will get called once for each message
    public int partition(String topic, Object key, byte[] keyBytes, Object value, byte[] valueBytes, Cluster cluster) {
        String countryName = ((String) value).split(":")[0];

        if (countryToPartitionMap.containsKey(countryName))
        {
            //If the country is mapped to particular partition return it
            return countryToPartitionMap.get(countryName);
        } else {
            //If no country is mapped to particular partition distribute between remaining partitions
            int noOfPartitions = cluster.topics().size();
            return value.hashCode() % noOfPartitions + countryToPartitionMap.size();
        }
    }

    @Override
    public void close() {
    }
}
```

## CustomProducer.java

- Set the Custom Partitioner class in Producer properties

```
configProperties.put(ProducerConfig.PARTITIONER_CLASS_CONFIG, CountryPartitioner.class.getCanonicalName());
```

```
vim /home/hadoop/Admatic/Kafka/java/Kafka-Producer/src/main/java/com/admatic/CustomProducer.java
```

```
package com.admatic;

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

import java.util.Properties;
import java.util.Scanner;

public class CustomProducer {

    public static void main(String[] argv) {
        if (argv.length != 1) {
            System.err.println("Usage: <topic>");
            return;
        }
        String topicName = argv[0];

        Scanner in = new Scanner(System.in);
        System.out.println("Enter message: ");

        Properties configProperties = new Properties();
        configProperties.put(ProducerConfig.BootstrapServersConfig, "localhost:9092");
        configProperties.put(ProducerConfig.KeySerializerClassConfig, "org.apache.kafka.common.serialization.ByteArraySerializer");
        configProperties.put(ProducerConfig.ValueSerializerClassConfig, "org.apache.kafka.common.serialization.StringSerializer");

        configProperties.put(ProducerConfig.PartitionerClassConfig, CountryPartitioner.class.getCanonicalName());
        configProperties.put("partitions.0", "India");
        configProperties.put("partitions.1", "USA");

        KafkaProducer<String, String> producer = new KafkaProducer<>(configProperties);
        String line = in.nextLine();
        while (!line.equals("exit")) {
            ProducerRecord<String, String> producerRecord = new ProducerRecord<>(topicName, line);
            producer.send(producerRecord);

            line = in.nextLine();
        }
        in.close();
        producer.close();
    }
}
```

## Create test-topic

```
kafka-topics.sh --delete --zookeeper localhost:2181 --topic test-topic
```

Topic **test**-topic is marked for deletion.

Note: This will have no impact if delete.topic.enable is not set to true.

```
kafka-topics.sh --create --zookeeper localhost:2181 --topic test-topic --partitions 5 --replication-factor 2
```

Created topic **"test-topic"**.

## Compile and Run

```
cd /home/hadoop/Admatic/Kafka/java/Kafka-Producer/
mvn compile
```

```
mvn exec:java -Dexec.mainClass="com.admatic.CustomProducer" -Dexec.args="test-topic"
```

```
India:hello
India:world
India:admatic
USA:United
USA:States
USA:America
UK:United Kingdom
Canada:Can
Earth:Mars
```

```
kafkacat -b localhost:9092 -t test-topic -f 'partition: %p, offset: %o, key: %k, value: %s\n'
```

```
partition: 0, offset: 0, key: , value: India:hello
partition: 0, offset: 1, key: , value: India:world
partition: 0, offset: 2, key: , value: India:admatic
partition: 1, offset: 0, key: , value: USA:United
partition: 1, offset: 1, key: , value: USA:States
partition: 1, offset: 2, key: , value: USA:America
partition: 2, offset: 0, key: , value: UK:United Kingdom
partition: 2, offset: 1, key: , value: Canada:Can
partition: 2, offset: 2, key: , value: Earth:Mars
```