

WeCanDoNow HandsOn Lab - Kafka Consumer Java Client

by Saravanan Sundaramoorthy

Kafka Consumer - Create a Project from Maven Template

```
sudo apt install maven -y
```

```
mvn archetype:generate -DgroupId=com.wecandonow -DartifactId=WeCanDoNow-Kafka-Consumer -DarchetypeArtifactId=maven-archetype-quickstart  
cd WeCanDoNow-Kafka-Consumer/
```

Add Kafka dependencies in pom.xml

```
cd WeCanDoNow-Kafka-Consumer
```

```
vim pom.xml
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">  
    <modelVersion>4.0.0</modelVersion>  
    <groupId>com.wecandonow</groupId>  
    <artifactId>Kafka-Producer</artifactId>  
    <packaging>jar</packaging>  
    <version>1.0-SNAPSHOT</version>  
    <name>Kafka-Producer</name>  
    <url>http://maven.apache.org</url>  
  
    <build>  
        <plugins>  
            <plugin>  
                <groupId>org.apache.maven.plugins</groupId>  
                <artifactId>maven-compiler-plugin</artifactId>  
                <version>3.7.0</version>  
                <configuration>  
                    <source>1.8</source>  
                    <target>1.8</target>  
                </configuration>  
            </plugin>  
        </plugins>  
    </build>  
</project>
```

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-assembly-plugin</artifactId>
  <version>2.4.1</version>
  <configuration>
    <archive>
      <manifest>
        <mainClass>com.wecandonow.App</mainClass>
      </manifest>
    </archive>
    <descriptorRefs>
      <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
    <skipAssembly>>false</skipAssembly>
  </configuration>
  <executions>
    <execution>
      <id>package</id>
      <phase>package</phase>
      <goals>
        <goal>single</goal>
      </goals>
    </execution>
  </executions>
</plugin>
</plugins>
</build>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka_2.12</artifactId>
    <version>1.0.1</version>
  </dependency>

  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-clients</artifactId>
    <version>1.0.1</version>
  </dependency>

  <dependency>
    <groupId>org.apache.kafka</groupId>
    <artifactId>kafka-streams</artifactId>
    <version>1.0.1</version>
  </dependency>

```

```
</dependencies>
</project>
```

Consumer API

The Consumer API allows applications to read streams of data from topics in the Kafka cluster.

Consumer.java

```
cd WeCanDoNow-Kafka-Consumer
vim src/main/java/com/wecandonow/Consumer.java
```

```
package com.wecandonow;

import org.apache.kafka.clients.consumer.ConsumerRecord;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.KafkaConsumer;

import java.util.Arrays;
import java.util.Properties;
import java.util.regex.Pattern;

public class Consumer
{
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Usage: <topic>");
            return;
        }

        Properties props = new Properties();
        props.put("bootstrap.servers", "localhost:9092");
        props.put("group.id", "wecandonow-group");
        props.put("enable.auto.commit", "true");
        props.put("auto.commit.interval.ms", "1000");
        props.put("key.deserializer", "org.apache.kafka.common.serialization.StringD
eserializer");
        props.put("value.deserializer", "org.apache.kafka.common.serialization.Strin
gDeserializer");

        KafkaConsumer<String, String> consumer = new KafkaConsumer<>(props);
        consumer.subscribe(Arrays.asList(args[0]));

        while (true) {
            ConsumerRecords<String, String> records = consumer.poll(100);
            for (ConsumerRecord<String, String> record : records)
                System.out.printf("partition = %d, offset = %d, key = %s, value = %s
%n", record.partition(), record.offset(), record.key(), record.value());
        }
    }
}
```

```
}  
}
```

Create wecandonow-topic

```
kafka-topics.sh --delete --zookeeper localhost:2181 --topic wecandonow-topic  
kafka-topics.sh --create --zookeeper localhost:2181 --topic wecandonow-topic --parti  
tions 5 --replication-factor 2
```

Created topic "wecandonow-topic".

Compile and Run the App

```
cd WeCanDoNow-Kafka-Consumer/
```

```
mvn compile  
mvn exec:java -Dexec.mainClass="com.wecandonow.Consumer" -Dexec.args="wecandonow-top  
ic"
```

Output

Without key

```
kafka-console-producer.sh --broker-list localhost:9092 --topic wecandonow-topic  
a  
b  
c  
admatic  
d  
e  
f  
wecandonow
```

```
partition = 4, offset = 0, key = null, value = a  
partition = 1, offset = 0, key = null, value = b  
partition = 3, offset = 0, key = null, value = c  
partition = 0, offset = 0, key = null, value = admatic  
partition = 2, offset = 0, key = null, value = d  
partition = 4, offset = 1, key = null, value = e  
partition = 1, offset = 1, key = null, value = f  
partition = 3, offset = 1, key = null, value = wecandonow
```

```
ubuntu@ip-172-31-42-149:~/WeCanDoNow-Kafka-Consumer$ mvn exec:java -Dexec.mainClass=  
"com.wecandonow.Consumer" -Dexec.args="wecandonow-topic"
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.wecandonow:Kafka-Producer >-----
[INFO] Building Kafka-Producer 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Kafka-Producer ---
log4j:WARN No appenders could be found for logger (org.apache.kafka.clients.consumer.ConsumerConfig).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
partition = 4, offset = 0, key = null, value = hello
partition = 1, offset = 0, key = null, value = folks
partition = 4, offset = 1, key = null, value = 4
partition = 0, offset = 0, key = null, value = 2
partition = 3, offset = 0, key = null, value = 3
partition = 2, offset = 0, key = null, value = 1
partition = 1, offset = 1, key = null, value = 5
partition = 0, offset = 1, key = null, value = 7
partition = 2, offset = 1, key = null, value = 6
partition = 3, offset = 1, key = null, value = 8
partition = 4, offset = 2, key = null, value =
partition = 4, offset = 3, key = null, value = e
partition = 3, offset = 2, key = null, value = d
partition = 3, offset = 3, key = null, value = i
partition = 4, offset = 4, key = null, value = j
```

```
3>
>4
>5
>6
>7
>8
>
>a
>b
>c
>d
>e
>f
>g
>h
>i
>j
>
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.wecandonow:Kafka-Producer >-----
[INFO] Building Kafka-Producer 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.1.0:java (default-cli) @ Kafka-Producer ---
log4j:WARN No appenders could be found for logger (org.apache.kafka.clients.consumer
.ConsumerConfig).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
partition = 1, offset = 2, key = null, value = a
partition = 2, offset = 2, key = null, value = b
partition = 0, offset = 2, key = null, value = c
partition = 1, offset = 3, key = null, value = f
partition = 2, offset = 3, key = null, value = g
partition = 0, offset = 3, key = null, value = h
```