**CASE STUDY**

**KAFKA**

**Pallav Gupta**

**46022374**

**CASE STUDY PROBLEM STATEMENT**

## Read the csv using spark

1. Produce first 10 lines of csv using Kafka producer (read first 10 lines and pass it to producer)
2. Consume the produced lines using Kafka consumer
3. Set the timer to 10sec, after 10 seconds next 10 records should be consumed and so on.

**SOLUTION**

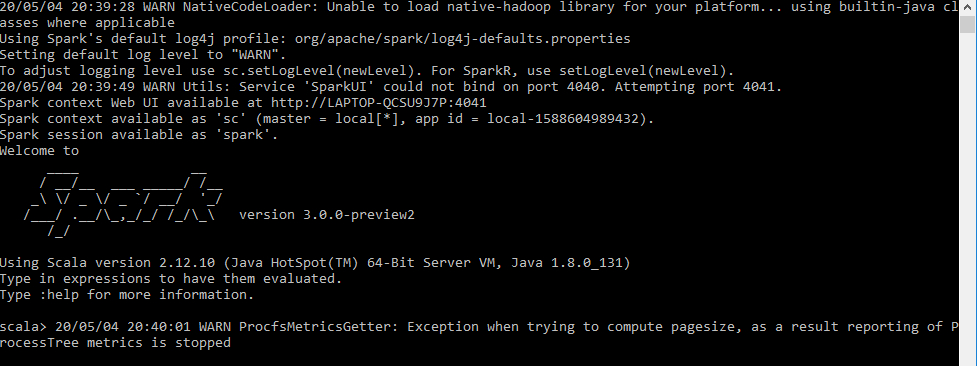
**DATA FILE USED -- User.csv**

**SOFTWARE USED**

* Intelli J
* Apache maven
* Kafka
* JDK 1.8

**SPARK VERSION**

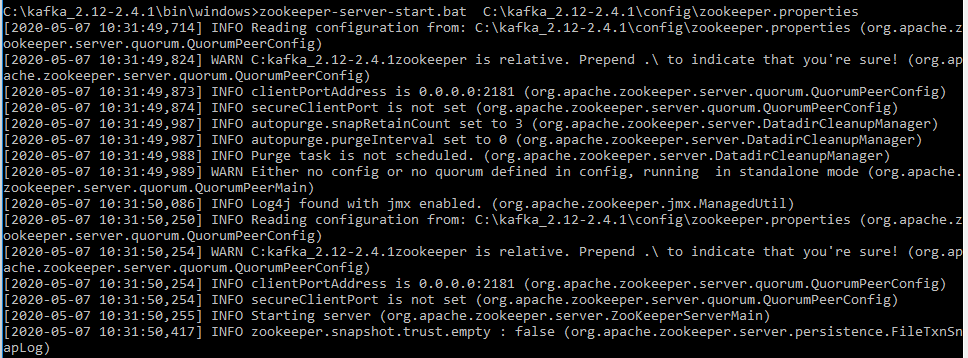
* Spark 3.0.0-preview2 version
* Scala 2.12.10 version



**STARTING ZOOKEEPER**

**COMMANDS:**

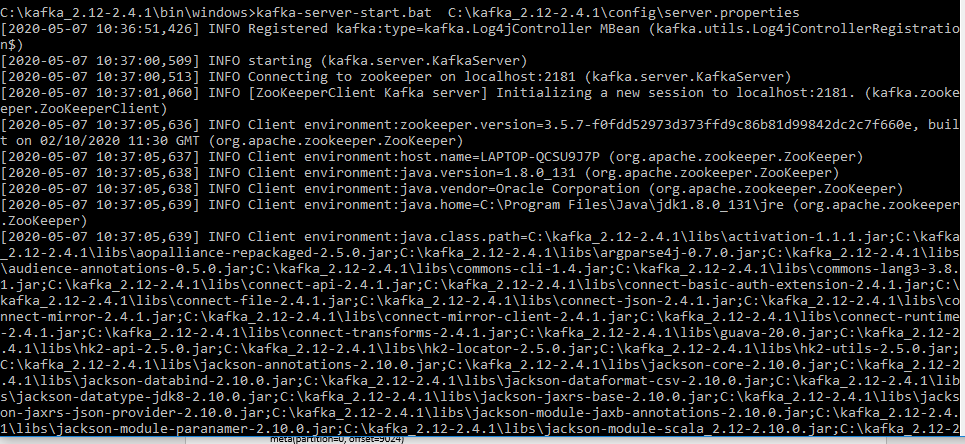
zookeeper-server-start.bat C:\kafka\_2.12-2.4.1\config\zookeeper.properties



**STARTING KAFKA**

**COMMANDS:**

kafka-server-start.bat C:\kafka\_2.12-2.4.1\config\server.properties

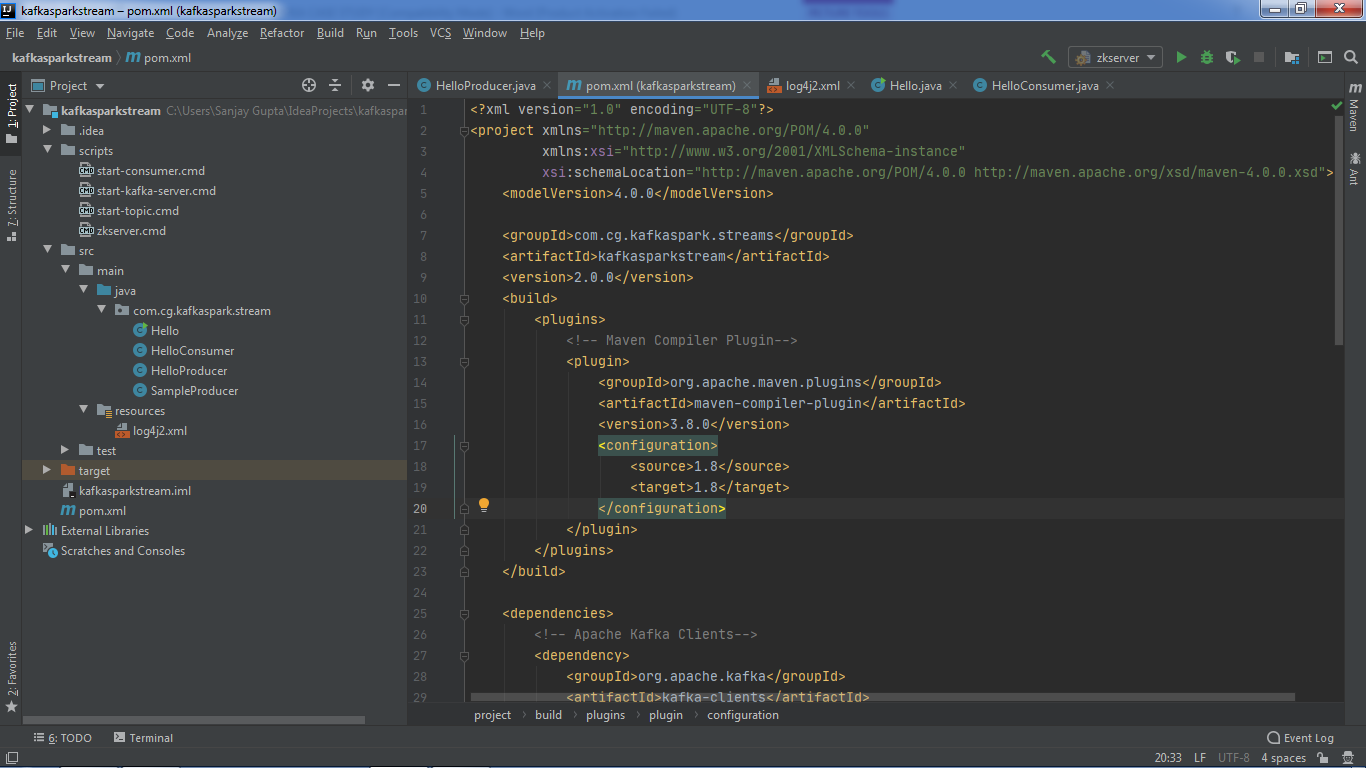


**POM.xml**

Load spark dependencies for maven

# CODE

<?xml version="1.0" encoding="UTF-8"?>  
<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/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.cg.kafkaspark.streams</groupId>  
 <artifactId>kafkasparkstream</artifactId>  
 <version>2.0.0</version>  
 <build>  
 <plugins>  
 <!-- Maven Compiler Plugin-->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.0</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
  
 <dependencies>  
 <!-- Apache Kafka Clients-->  
 <dependency>  
 <groupId>org.apache.kafka</groupId>  
 <artifactId>kafka-clients</artifactId>  
 <version>2.0.0</version>  
 </dependency>  
 <!-- Apache Kafka Streams-->  
 <dependency>  
 <groupId>org.apache.kafka</groupId>  
 <artifactId>kafka-streams</artifactId>  
 <version>2.0.0</version>  
 </dependency>  
 <!-- Apache Log4J2 binding for SLF4J -->  
 <dependency>  
 <groupId>org.apache.logging.log4j</groupId>  
 <artifactId>log4j-slf4j-impl</artifactId>  
 <version>2.11.0</version>  
 </dependency>  
 <!-- JUnit5 Jupiter -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.3.1</version>  
 <scope>test</scope>  
 </dependency>  
 <!-- JUnit 5 Jupiter Engine -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-engine</artifactId>  
 <version>5.3.1</version>  
 <scope>test</scope>  
 </dependency>  
 <!-- JUnit 5 Jupiter Parameterized Testing -->  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-params</artifactId>  
 <version>5.3.1</version>  
 <scope>test</scope>  
 </dependency>  
 </dependencies>  
  
</project>



**CONSUMER CLASS**

**CODE**

package com.cg.kafkaspark.stream;  
  
import org.apache.kafka.clients.consumer.ConsumerConfig;  
import org.apache.kafka.clients.consumer.ConsumerRecord;  
import org.apache.kafka.clients.consumer.ConsumerRecords;  
import org.apache.kafka.clients.consumer.KafkaConsumer;  
import org.apache.kafka.common.serialization.StringDeserializer;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.time.Duration;  
import java.util.Arrays;  
import java.util.Collections;  
import java.util.Properties;  
  
public class HelloConsumer {  
 public HelloConsumer(String tname) {  
 Logger logger= LoggerFactory.*getLogger*(HelloConsumer.class.getName());  
 String bootstrapServers="127.0.0.1:9092";  
 String grp\_id="third\_app";  
 String topic=tname;  
 //Creating consumer properties  
 Properties properties=new Properties();  
 properties.setProperty(ConsumerConfig.*BOOTSTRAP\_SERVERS\_CONFIG*,bootstrapServers);  
 properties.setProperty(ConsumerConfig.*KEY\_DESERIALIZER\_CLASS\_CONFIG*, StringDeserializer.class.getName());  
 properties.setProperty(ConsumerConfig.*VALUE\_DESERIALIZER\_CLASS\_CONFIG*,StringDeserializer.class.getName());  
 properties.setProperty(ConsumerConfig.*GROUP\_ID\_CONFIG*,grp\_id);  
 properties.setProperty(ConsumerConfig.*AUTO\_OFFSET\_RESET\_CONFIG*,"earliest");  
 //creating consumer  
 KafkaConsumer<String,String> consumer= new KafkaConsumer<String,String>(properties);  
 //Subscribing  
 consumer.subscribe(Arrays.*asList*(topic));  
 //polling  
 while(true){  
 ConsumerRecords<String,String> records=consumer.poll(Duration.*ofMillis*(100));  
 for(ConsumerRecord<String,String> record: records){  
 logger.info(record.value());  
// logger.info("Key: "+ record.key() + ", Value:" +record.value());  
// logger.info("Partition:" + record.partition()+",Offset:"+record.offset());  
 }  
  
  
 }  
 }  
}

**PRODUCER CLASS**

**CODE**

package com.cg.kafkaspark.stream;  
  
import org.apache.kafka.clients.producer.KafkaProducer;  
import org.apache.kafka.clients.producer.ProducerConfig;  
import org.apache.kafka.clients.producer.ProducerRecord;  
import org.apache.kafka.common.KafkaException;  
import org.apache.kafka.common.serialization.IntegerSerializer;  
import org.apache.kafka.common.serialization.StringSerializer;  
import org.apache.logging.log4j.LogManager;  
import org.apache.logging.log4j.Logger;  
  
import java.util.\*;  
import java.io.\*;  
public class HelloProducer {  
 private static final Logger logger = LogManager.getLogger(HelloProducer.class);  
  
 public HelloProducer(String tname,int numEvent) {  
 String topicName;  
 int numEvents;  
//  
 topicName = tname;  
 numEvents = numEvent;  
 logger.info("Starting HelloProducer...");  
 logger.debug("topicName=" + topicName + ", numEvents=" + numEvents);  
 logger.trace("Creating Kafka Producer...");  
 Properties props = new Properties();  
 props.put(ProducerConfig.CLIENT\_ID\_CONFIG, "HelloProducer");  
 props.put(ProducerConfig.BOOTSTRAP\_SERVERS\_CONFIG, "localhost:9092");  
 props.put(ProducerConfig.KEY\_SERIALIZER\_CLASS\_CONFIG, IntegerSerializer.class.getName());  
 props.put(ProducerConfig.VALUE\_SERIALIZER\_CLASS\_CONFIG, StringSerializer.class.getName());  
 KafkaProducer<Integer, String> producer = new KafkaProducer<>(props);  
 logger.trace("Start sending messages...");  
 try {  
 FileInputStream fis=new FileInputStream("C:\\Users\\Sanjay Gupta\\Desktop\\Capgemini\\kafka\\user.csv");  
 Scanner scanner=new Scanner(fis);  
 int i=1;  
 while(scanner.hasNextLine() && i<=numEvents){  
 producer.send(new ProducerRecord<>(topicName,i,scanner.nextLine()));  
 i++;  
 }  
 } catch (KafkaException e) {  
 logger.error("Exception occurred – Check log for more details.\n" + e.getMessage());  
 System.exit(-1);  
 }  
 catch(IOException e)  
 {  
 e.printStackTrace();  
 }  
 finally {  
 logger.info("Finished HelloProducer – Closing Kafka Producer.");  
 producer.close();  
 }  
 }  
}

**RUNNER CLASS**

**CODE**

package com.cg.kafkaspark.stream;  
  
public class Hello {  
 public static void main(String[] args) {  
 String topic="kafkacasestudy";  
 int numEvent=10;  
 HelloProducer hp=new HelloProducer(topic,numEvent);  
 HelloConsumer hc=new HelloConsumer(topic);  
 }  
}

**LOG4J2.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<Configuration status="ERROR">  
 <Appenders>  
 <Console name="stdout" target="SYSTEM\_OUT">  
 <PatternLayout pattern="[%d] (%c) - %p %m %n"/>  
 </Console>  
 </Appenders>  
 <Loggers>  
 <Root level="error">  
 <AppenderRef ref="stdout"/>  
 </Root>  
 <Logger name="org.apache.kafka.clients" level="warn" additivity="false">  
 <AppenderRef ref="stdout"/>  
 </Logger>  
 <Logger name="com.cg.kafkaspark.stream" level="trace" additivity="false">  
 <AppenderRef ref="stdout"/>  
 </Logger>  
 </Loggers>  
</Configuration>

**OUTPUT**

