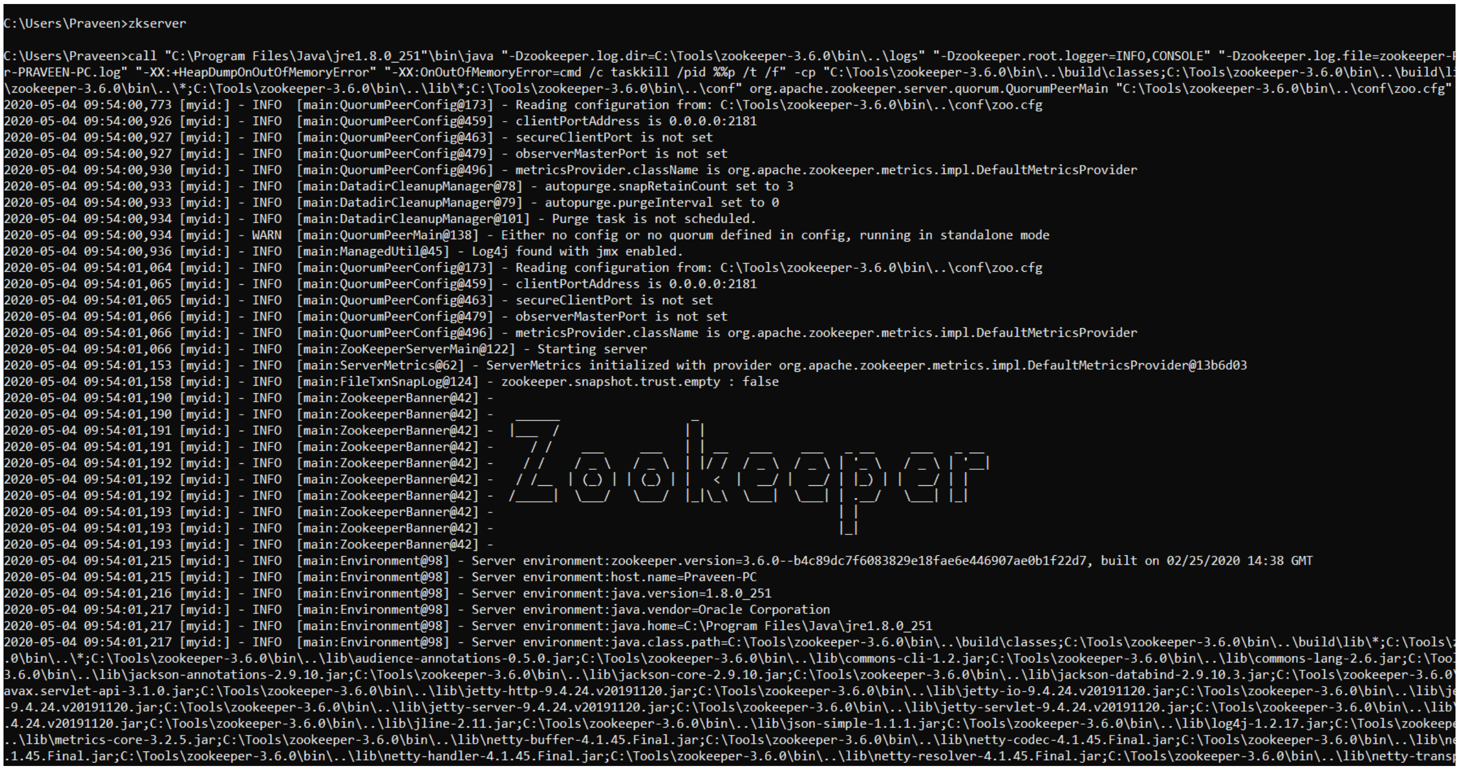
Kafka Streaming in Java

Objective: To send desired data from a CSV file via Java as a Producer for Kafka, which will be consumed by Java Consumer

STEPS:

Step 1: Start Zookeeper

Command: ‘zkserver’

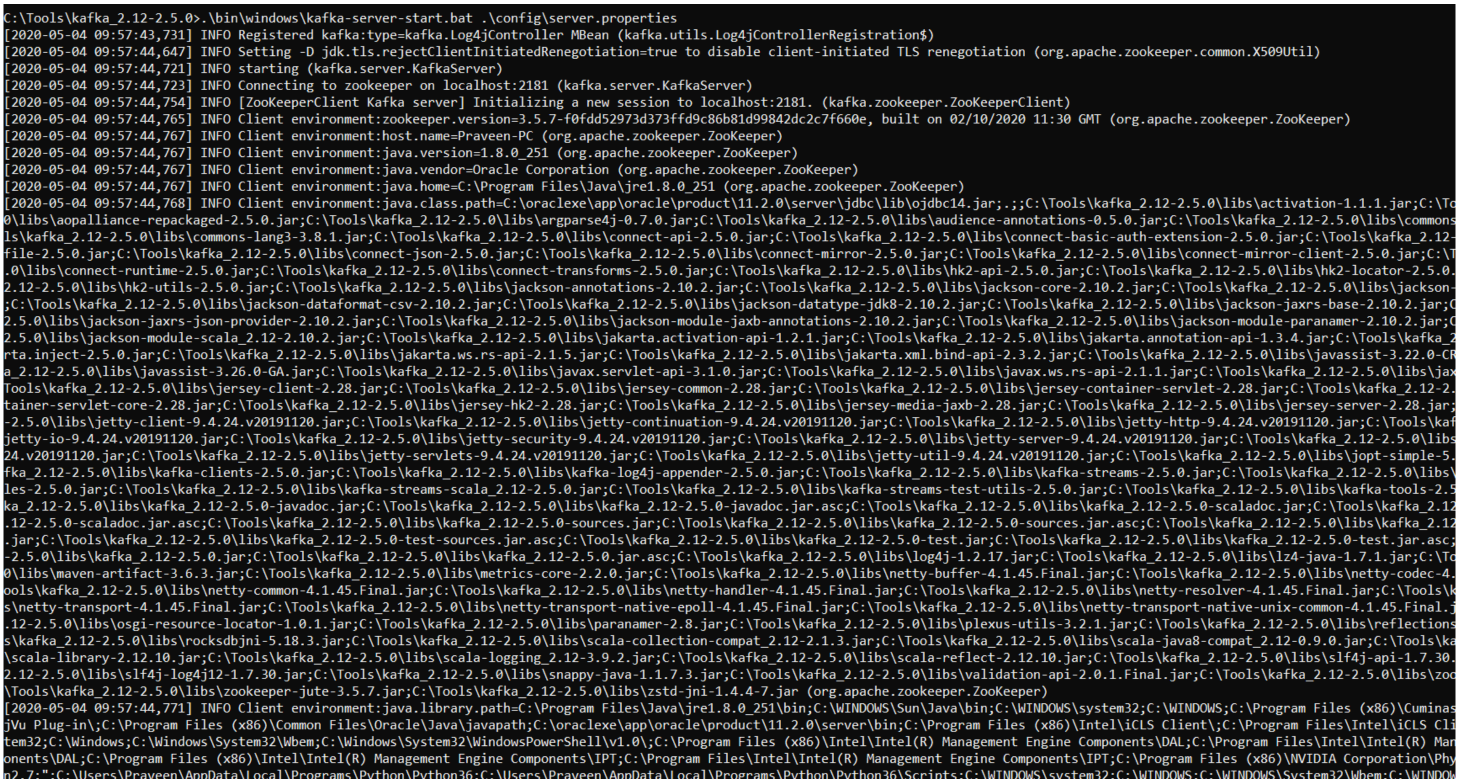


```
C:\Users\Praveen>zkserver

C:\Users\Praveen>call "C:\Program Files\Java\jre1.8.0_251"\bin\java "-Dzookeeper.log.dir=C:\Tools\zookeeper-3.6.0\bin\..\logs" "-Dzookeeper.root.logger=INFO,CONSOLE" "-Dzookeeper.log.file=zookeeper-P
r-PRAVEEN-PC.log" "-XX:+HeapDumpOnOutOfMemoryError" "-XX:OnOutOfMemoryError=cmd /c taskkill /pid %%p /t /f" -cp "C:\Tools\zookeeper-3.6.0\bin\..\build\classes;C:\Tools\zookeeper-3.6.0\bin\..\build\li
\zookeeper-3.6.0\bin\..\*;C:\Tools\zookeeper-3.6.0\bin\..\lib\*;C:\Tools\zookeeper-3.6.0\bin\..\conf" org.apache.zookeeper.server.quorum.QuorumPeerMain "C:\Tools\zookeeper-3.6.0\bin\..\conf\zoo.cfg"
2020-05-04 09:54:00,773 [myid:] - INFO  [main:QuorumPeerConfig@173] - Reading configuration from: C:\Tools\zookeeper-3.6.0\bin\..\conf\zoo.cfg
2020-05-04 09:54:00,926 [myid:] - INFO  [main:QuorumPeerConfig@459] - clientPortAddress is 0.0.0.0:2181
2020-05-04 09:54:00,927 [myid:] - INFO  [main:QuorumPeerConfig@463] - secureClientPort is not set
2020-05-04 09:54:00,927 [myid:] - INFO  [main:QuorumPeerConfig@479] - observerMasterPort is not set
2020-05-04 09:54:00,930 [myid:] - INFO  [main:QuorumPeerConfig@496] - metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider
2020-05-04 09:54:00,933 [myid:] - INFO  [main:DatadirCleanupManager@78] - autopurge.snapRetainCount set to 3
2020-05-04 09:54:00,933 [myid:] - INFO  [main:DatadirCleanupManager@79] - autopurge.purgeInterval set to 0
2020-05-04 09:54:00,934 [myid:] - INFO  [main:DatadirCleanupManager@101] - Purge task is not scheduled.
2020-05-04 09:54:00,934 [myid:] - WARN  [main:QuorumPeerMain@138] - Either no config or no quorum defined in config, running in standalone mode
2020-05-04 09:54:00,936 [myid:] - INFO  [main:ManagedUtil@45] - Log4j found with jmx enabled.
2020-05-04 09:54:01,064 [myid:] - INFO  [main:QuorumPeerConfig@173] - Reading configuration from: C:\Tools\zookeeper-3.6.0\bin\..\conf\zoo.cfg
2020-05-04 09:54:01,065 [myid:] - INFO  [main:QuorumPeerConfig@459] - clientPortAddress is 0.0.0.0:2181
2020-05-04 09:54:01,065 [myid:] - INFO  [main:QuorumPeerConfig@463] - secureClientPort is not set
2020-05-04 09:54:01,066 [myid:] - INFO  [main:QuorumPeerConfig@479] - observerMasterPort is not set
2020-05-04 09:54:01,066 [myid:] - INFO  [main:QuorumPeerConfig@496] - metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider
2020-05-04 09:54:01,066 [myid:] - INFO  [main:ZooKeeperServerMain@122] - Starting server
2020-05-04 09:54:01,153 [myid:] - INFO  [main:ServerMetrics@62] - ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@13b6d03
2020-05-04 09:54:01,158 [myid:] - INFO  [main:FileTxnSnapLog@124] - zookeeper.snapshot.trust.empty : false
2020-05-04 09:54:01,190 [myid:] - INFO  [main:ZookeeperBanner@42] -
2020-05-04 09:54:01,190 [myid:] - INFO  [main:ZookeeperBanner@42] -   ______                  _
2020-05-04 09:54:01,191 [myid:] - INFO  [main:ZookeeperBanner@42] -  |___  /                 | |
2020-05-04 09:54:01,191 [myid:] - INFO  [main:ZookeeperBanner@42] -     / /    ___     ___   | | __   ___    ___   _ __     ___   _ __
2020-05-04 09:54:01,192 [myid:] - INFO  [main:ZookeeperBanner@42] -    / /    / _ \   / _ \  | |/ /  / _ \  / _ \ | '_ \   / _ \ | '__|
2020-05-04 09:54:01,192 [myid:] - INFO  [main:ZookeeperBanner@42] -   / /__  | (_) | | (_) | |   <  |  __/ |  __/ | |_) | |  __/ | |
2020-05-04 09:54:01,192 [myid:] - INFO  [main:ZookeeperBanner@42] -  /_____|  \___/   \___/  |_|\_\  \___|  \___| | .__/   \___| |_|
2020-05-04 09:54:01,193 [myid:] - INFO  [main:ZookeeperBanner@42] -                                               | |
2020-05-04 09:54:01,193 [myid:] - INFO  [main:ZookeeperBanner@42] -                                               |_|
2020-05-04 09:54:01,193 [myid:] - INFO  [main:ZookeeperBanner@42] -
2020-05-04 09:54:01,215 [myid:] - INFO  [main:Environment@98] - Server environment:zookeeper.version=3.6.0--b4c89dc7f6083829e18fae6e446907ae0b1f22d7, built on 02/25/2020 14:38 GMT
2020-05-04 09:54:01,215 [myid:] - INFO  [main:Environment@98] - Server environment:host.name=Praveen-PC
2020-05-04 09:54:01,216 [myid:] - INFO  [main:Environment@98] - Server environment:java.version=1.8.0_251
2020-05-04 09:54:01,217 [myid:] - INFO  [main:Environment@98] - Server environment:java.vendor=Oracle Corporation
2020-05-04 09:54:01,217 [myid:] - INFO  [main:Environment@98] - Server environment:java.home=C:\Program Files\Java\jre1.8.0_251
2020-05-04 09:54:01,217 [myid:] - INFO  [main:Environment@98] - Server environment:java.class.path=C:\Tools\zookeeper-3.6.0\bin\..\build\classes;C:\Tools\zookeeper-3.6.0\bin\..\build\lib\*;C:\Tools\z
.0\bin\..\*;C:\Tools\zookeeper-3.6.0\bin\..\lib\audience-annotations-0.5.0.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\commons-cli-1.2.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\commons-lang-2.6.jar;C:\Tool
3.6.0\bin\..\lib\jackson-annotations-2.9.10.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jackson-core-2.9.10.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jackson-databind-2.9.10.3.jar;C:\Tools\zookeeper-3.6.0\
avax.servlet-api-3.1.0.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jetty-http-9.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jetty-io-9.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\je
-9.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jetty-server-9.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jetty-servlet-9.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\
.4.24.v20191120.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\jline-2.11.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\json-simple-1.1.1.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\log4j-1.2.17.jar;C:\Tools\zookeepe
..\lib\metrics-core-3.2.5.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\netty-buffer-4.1.45.Final.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\netty-codec-4.1.45.Final.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\ne
.1.45.Final.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\netty-handler-4.1.45.Final.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\netty-resolver-4.1.45.Final.jar;C:\Tools\zookeeper-3.6.0\bin\..\lib\netty-transp
```

Step 2: To Start Kafka Server

Command: “ .\bin\windows\kafka-server-start.bat .\config\server.properties”

Note: Change your directory while entering command to Kafka folder



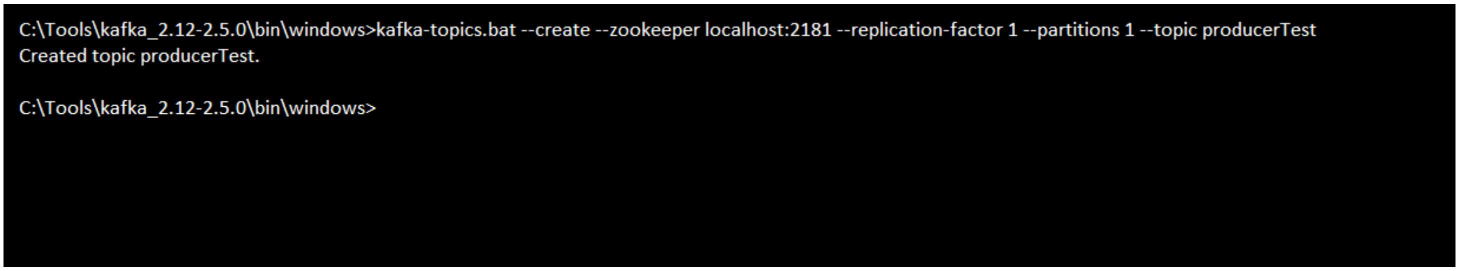
```
C:\Tools\kafka_2.12-2.5.0>.\bin\windows\kafka-server-start.bat .\config\server.properties
[2020-05-04 09:57:43,731] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2020-05-04 09:57:44,647] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2020-05-04 09:57:44,721] INFO starting (kafka.server.KafkaServer)
[2020-05-04 09:57:44,723] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2020-05-04 09:57:44,754] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2020-05-04 09:57:44,765] INFO Client environment:zookeeper.version=3.5.7-f0fdd52973d373ffd9c86b81d99842dc2c7f660e, built on 02/10/2020 11:30 GMT (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,767] INFO Client environment:host.name=Praveen-PC (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,767] INFO Client environment:java.version=1.8.0_251 (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,767] INFO Client environment:java.vendor=Oracle Corporation (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,767] INFO Client environment:java.home=C:\Program Files\Java\jre1.8.0_251 (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,768] INFO Client environment:java.class.path=C:\oraclexe\app\oracle\product\11.2.0\server\jdbc\lib\ojdbc14.jar;.;;C:\Tools\kafka_2.12-2.5.0\libs\activation-1.1.1.jar;C:\To
0\libs\aopalliance-repackaged-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\argparse4j-0.7.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\audience-annotations-0.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\commons
ls\kafka_2.12-2.5.0\libs\commons-lang3-3.8.1.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-api-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-basic-auth-extension-2.5.0.jar;C:\Tools\kafka_2.12-
file-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-json-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-mirror-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-mirror-client-2.5.0.jar;C:\T
.0\libs\connect-runtime-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\connect-transforms-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\hk2-api-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\hk2-locator-2.5.0.
2.12-2.5.0\libs\hk2-utils-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-annotations-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-core-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-
;C:\Tools\kafka_2.12-2.5.0\libs\jackson-dataformat-csv-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-datatype-jdk8-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-jaxrs-base-2.10.2.jar;C
2.5.0\libs\jackson-jaxrs-json-provider-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-module-jaxb-annotations-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jackson-module-paranamer-2.10.2.jar;C
2.5.0\libs\jackson-module-scala_2.12-2.10.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\jakarta.activation-api-1.2.1.jar;C:\Tools\kafka_2.12-2.5.0\libs\jakarta.annotation-api-1.3.4.jar;C:\Tools\kafka_2
rta.inject-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\jakarta.ws.rs-api-2.1.5.jar;C:\Tools\kafka_2.12-2.5.0\libs\jakarta.xml.bind-api-2.3.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\javassist-3.22.0-CR
a_2.12-2.5.0\libs\javassist-3.26.0-GA.jar;C:\Tools\kafka_2.12-2.5.0\libs\javax.servlet-api-3.1.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\javax.ws.rs-api-2.1.1.jar;C:\Tools\kafka_2.12-2.5.0\libs\jax
Tools\kafka_2.12-2.5.0\libs\jersey-client-2.28.jar;C:\Tools\kafka_2.12-2.5.0\libs\jersey-common-2.28.jar;C:\Tools\kafka_2.12-2.5.0\libs\jersey-container-servlet-2.28.jar;C:\Tools\kafka_2.12-2.
tainer-servlet-core-2.28.jar;C:\Tools\kafka_2.12-2.5.0\libs\jersey-hk2-2.28.jar;C:\Tools\kafka_2.12-2.5.0\libs\jersey-media-jaxb-2.28.jar;C:\Tools\kafka_2.12-2.5.0\libs\jersey-server-2.28.jar;
-2.5.0\libs\jetty-client-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-continuation-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-http-9.4.24.v20191120.jar;C:\Tools\kaf
jetty-io-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-security-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-server-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs
24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-servlets-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jetty-util-9.4.24.v20191120.jar;C:\Tools\kafka_2.12-2.5.0\libs\jopt-simple-5.
fka_2.12-2.5.0\libs\kafka-clients-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka-log4j-appender-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka-streams-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\
les-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka-streams-scala_2.12-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka-streams-test-utils-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka-tools-2.5
ka_2.12-2.5.0\libs\kafka_2.12-2.5.0-javadoc.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-javadoc.jar.asc;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-scaladoc.jar;C:\Tools\kafka_2.12
.12-2.5.0-scaladoc.jar.asc;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-sources.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-sources.jar.asc;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12
.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-test-sources.jar.asc;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-test.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0-test.jar.asc;
-2.5.0\libs\kafka_2.12-2.5.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\kafka_2.12-2.5.0.jar.asc;C:\Tools\kafka_2.12-2.5.0\libs\log4j-1.2.17.jar;C:\Tools\kafka_2.12-2.5.0\libs\lz4-java-1.7.1.jar;C:\To
0\libs\maven-artifact-3.6.3.jar;C:\Tools\kafka_2.12-2.5.0\libs\metrics-core-2.2.0.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-buffer-4.1.45.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-codec-4.
ools\kafka_2.12-2.5.0\libs\netty-common-4.1.45.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-handler-4.1.45.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-resolver-4.1.45.Final.jar;C:\Tools\k
s\netty-transport-4.1.45.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-transport-native-epoll-4.1.45.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\netty-transport-native-unix-common-4.1.45.Final.j
.12-2.5.0\libs\osgi-resource-locator-1.0.1.jar;C:\Tools\kafka_2.12-2.5.0\libs\paranamer-2.8.jar;C:\Tools\kafka_2.12-2.5.0\libs\plexus-utils-3.2.1.jar;C:\Tools\kafka_2.12-2.5.0\libs\reflections
s\kafka_2.12-2.5.0\libs\rocksdbjni-5.18.3.jar;C:\Tools\kafka_2.12-2.5.0\libs\scala-collection-compat_2.12-2.1.3.jar;C:\Tools\kafka_2.12-2.5.0\libs\scala-java8-compat_2.12-0.9.0.jar;C:\Tools\ka
\scala-library-2.12.10.jar;C:\Tools\kafka_2.12-2.5.0\libs\scala-logging_2.12-3.9.2.jar;C:\Tools\kafka_2.12-2.5.0\libs\scala-reflect-2.12.10.jar;C:\Tools\kafka_2.12-2.5.0\libs\slf4j-api-1.7.30.
2.12-2.5.0\libs\slf4j-log4j12-1.7.30.jar;C:\Tools\kafka_2.12-2.5.0\libs\snappy-java-1.1.7.3.jar;C:\Tools\kafka_2.12-2.5.0\libs\validation-api-2.0.1.Final.jar;C:\Tools\kafka_2.12-2.5.0\libs\zoo
\Tools\kafka_2.12-2.5.0\libs\zookeeper-jute-3.5.7.jar;C:\Tools\kafka_2.12-2.5.0\libs\zstd-jni-1.4.4-7.jar (org.apache.zookeeper.ZooKeeper)
[2020-05-04 09:57:44,771] INFO Client environment:java.library.path=C:\Program Files\Java\jre1.8.0_251\bin;C:\WINDOWS\Sun\Java\bin;C:\WINDOWS\system32;C:\WINDOWS;C:\Program Files (x86)\Cuminas
jVu Plug-in\;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\oraclexe\app\oracle\product\11.2.0\server\bin;C:\Program Files (x86)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Cli
tem32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Man
onents\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\NVIDIA Corporation\Phy
n2.7;";C:\Users\Praveen\AppData\Local\Programs\Python\Python36;C:\Users\Praveen\AppData\Local\Programs\Python\Python36\Scripts;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOW
```

Step 3: Creating a Topic for Kafka to which messages will be send and retrieve

Command: “kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic producerTest”

“producerTest” is the name of our Topic

Note: Change your directory to /bin/windows while entering the command



```
C:\Tools\kafka_2.12-2.5.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic producerTest
Created topic producerTest.

C:\Tools\kafka_2.12-2.5.0\bin\windows>
```

The default port number for zookeeper: 2181

Replication factor: 1

Partition made: 1

Step 4: Creating a Java Producer

# Package Manager: Since we are using Java we will be using Maven as our package manager to download dependencies, using Intellij Idea

# Adding Java Kafka Client dependencies to .iml/.xml file. These dependencies are to be added in order to run the program

```
<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
  <component name="NewModuleRootManager" inherit-compiler-output="true">
    <exclude-output />
    <content url="file://$MODULE_DIR$">
      <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
    </content>
    <orderEntry type="inheritedJdk" />
    <orderEntry type="sourceFolder" forTests="false" />
    <orderEntry type="library" name="org.apache.kafka:kafka_2.12:0.10.1.1" level="project" />
    <orderEntry type="library" name="org.apache.kafka:connect-api:1.0.1" level="project" />
    <orderEntry type="library" name="com.opencsv:opencsv:5.1" level="project" />
  </component>
</module>
```

# Configuration needed to connect to Kafka Servers:

- We need some properties that will help us to connect to Kafka server.
- These properties will be key-value pair.

# bootstrap.servers:

- This property will define the address of the Kafka server.
- It is the combination of IP address and Port on which Kafka server is hosted.
- Default value: localhost:9092.

# key.serializer:

- Kafka stores the data as key-value pair in Topics.
- We need to define the type of serializer that will be needed to convert the content of key so that it could be sent to Kafka server.
- If key is of String type, then we need to set its value as org.apache.kafka.common.serialization.StringSerializer

# value.serializer:

- We need to define the type of serializer that will be needed to convert the content of value so that it could be sent to Kafka server.
- If value is of String type, then we need to set its value as org.apache.kafka.common.serialization.StringSerializer

```
Properties properties = new Properties();
properties.put("bootstrap.servers", "localhost:9092");
properties.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
properties.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
```

# Publish data to Kafka server:

- ProducerRecord type object is created which is send to the Kafka Server

```
ProducerRecord producerRecord = new ProducerRecord(topic_name, key, value)
```

- Topic name is where data will be published, data can be consumed from topic.
- Key inside which data is stored.
- Value is the actual data.

Note: We will avoid making a key in our program

Connection to Kafka and sending message:

- We need to make a connection to the Kafka server using KafkaProducer class.
- KafkaProducer takes Properties object as an argument to create a connection.

```
KafkaProducer kafkaProducer = new KafkaProducer(properties);
```

- We can send data to Kafka server by using send command.

```
kafkaProducer.send(new ProducerRecord("producerTest",finalData));
```

- producerTest is the name of Topic.
- finalData is the value of data to be published.

```
kafkaProducer.close();
```

- To close the connection, we created.

CSV:

Producer will publish the data to Kafka server by reading it from a CSV file:

```
String path = " kafka_data\\team_info.csv";
FileReader filereader = new FileReader(path);
CSVReader csvReader = new CSVReader(filereader);
String[] nextRecord;
```

- path is a String type variable containing the address of a CSV file.
- Opening CSV file by creating an object of FileReader class.
- Passing the FileReader object as an argument while creating CSVReader object.
- csvReader is the object containing the data of a CSV file.
- nextRecord is a String array which will contain one line of data at a time.

```
int counter = 0;
System.out.println("Enter the line number from which you want to publish data: ");
int lineNumStartFrom = sc.nextInt();
System.out.println("\nEnter the line number to which you want to publish data: ");
int lineNumTo = sc.nextInt();
while ((nextRecord = csvReader.readNext()) != null) {
   if (counter > lineNumStartFrom && counter <= lineNumTo) {
      String data = Arrays.toString(nextRecord);
      String finalData = data.substring(1, data.length()-1);
      kafkaProducer.send(new ProducerRecord("producerTest",finalData));
       Thread.sleep(1000);
   }
   counter++;
}
```

Final Code:

```
import java.util.Arrays;
import java.util.Scanner;
import java.io.FileReader;
import java.util.Properties;
import com.opencsv.CSVReader;
import org.apache.kafka.clients.producer.KafkaProducer;
import org.apache.kafka.clients.producer.ProducerRecord;

public class Producer {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      Properties properties = new Properties();
      properties.put("bootstrap.servers", "localhost:9092");
      properties.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
      properties.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
      KafkaProducer kafkaProducer = new KafkaProducer(properties);
      try {
         String path = " kafka_data\\team_info.csv";
         FileReader filereader = new FileReader(path);
         CSVReader csvReader = new CSVReader(filereader);
         String[] nextRecord;
         int counter = 0;
         System.out.println("Enter the line number from which you want to publish data: ");
         int lineNumStartFrom = sc.nextInt();
         System.out.println("\nEnter the line number to which you want to publish data: ");
         int lineNumTo = sc.nextInt();
         while ((nextRecord = csvReader.readNext()) != null) {
            if (counter > lineNumStartFrom && counter <= lineNumTo) {
               String data = Arrays.toString(nextRecord);
               String finalData = data.substring(1, data.length()-1);
               kafkaProducer.send(new ProducerRecord("producerTest",finalData));
               Thread.sleep(1000);
            }
            counter++;
         }
      }
      catch (Exception e) {
         e.printStackTrace();
      }
      finally {
         kafkaProducer.close();
      }
   }
}
```

Step 5: Creating a Java Consumer

# Package Manager: Since we are using Java we will be using Maven as our package manager to download dependencies, using Intellij Idea

# Adding Java Kafka Client dependencies to .iml/.xml file. These dependencies are to be added in order to run the program

```xml
<?xml version="1.0" encoding="UTF-8"?>
<module type="JAVA_MODULE" version="4">
 <component name="NewModuleRootManager" inherit-compiler-output="true">
  <exclude-output />
  <content url="file://$MODULE_DIR$">
   <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
  </content>
  <orderEntry type="inheritedJdk" />
  <orderEntry type="sourceFolder" forTests="false" />
  <orderEntry type="library" name="org.apache.kafka:connect-api:1.0.1" level="project" />
 </component>
</module>
```

# Connection Details:

- topic variable contains the name of Topic which we are subscribed to.
- bootstrap.servers, key.deserializer, value.deserializer are explained in producer.
- ‘enable.auto.commit’, ‘true’ means consumer will commit automatically on receiving messages.
- auto.commit.interval.ms = 1,000ms
- session.timeout.ms = 30,000ms

```
String topic = "producerTest";
String group = "producerTest";
Properties properties = new Properties();
properties.put("bootstrap.servers", "localhost:9092");
properties.put("group.id", group);
properties.put("enable.auto.commit", "true");
properties.put("auto.commit.interval.ms", "1000");
properties.put("session.timeout.ms", "30000");
properties.put("key.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");
properties.put("value.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");
```

# Connection to Kafka and sending message:

- We need to make a connection to the Kafka server using KafkaConsumer class.
- KafkaConsumer takes Properties object as an argument to create a connection.

```
KafkaConsumer<String, String> consumer = new KafkaConsumer<String, String>(properties);
```

# Subscribing to the Topic

```
consumer.subscribe(Arrays.asList(topic));
```

Printing messages received from selected Topic by subscribing it

```
while (true) {
   ConsumerRecords<String, String> records = consumer.poll(100);
   for (ConsumerRecord<String, String> record : records)
      System.out.printf(record.value()+"\n");
}
```

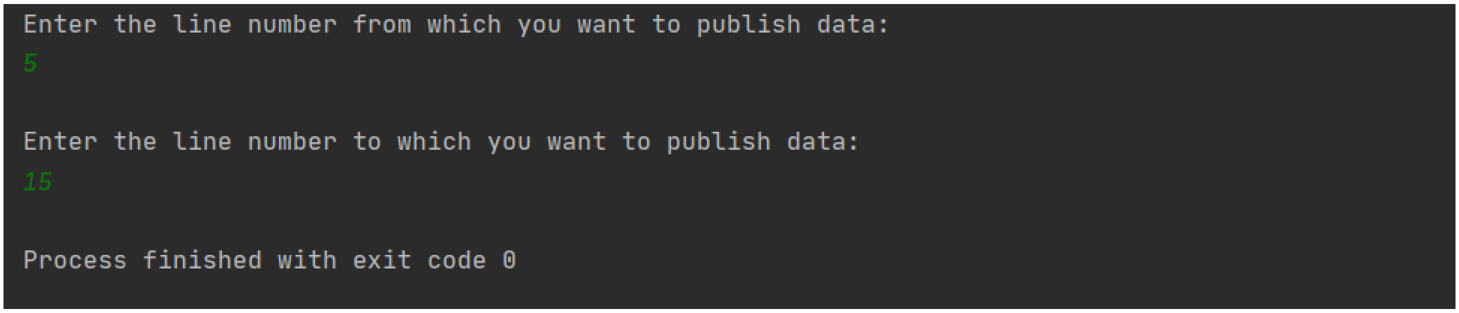
Final Code:

```
import java.util.*;
import org.apache.kafka.clients.consumer.KafkaConsumer;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.ConsumerRecord;

public class Consumer {
   public static void main(String[] args) throws Exception {

      String topic = "producerTest";
      String group = "producerTest";
      Properties properties = new Properties();
      properties.put("bootstrap.servers", "localhost:9092");
      properties.put("group.id", group);
      properties.put("enable.auto.commit", "true");
      properties.put("auto.commit.interval.ms", "1000");
      properties.put("session.timeout.ms", "30000");
      properties.put("key.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");
      properties.put("value.deserializer", "org.apache.kafka.common.serialization.StringDeserializer");
      KafkaConsumer<String, String> consumer = new KafkaConsumer<String, String>(properties);

      consumer.subscribe(Arrays.asList(topic));
      System.out.println("Subscribed to topic " + topic);

      while (true) {
         ConsumerRecords<String, String> records = consumer.poll(100);
         for (ConsumerRecord<String, String> record : records)
            System.out.printf(record.value()+"\n");
      }
   }
}
```

Final Output:

Java Output-IntelliJ (Producer):

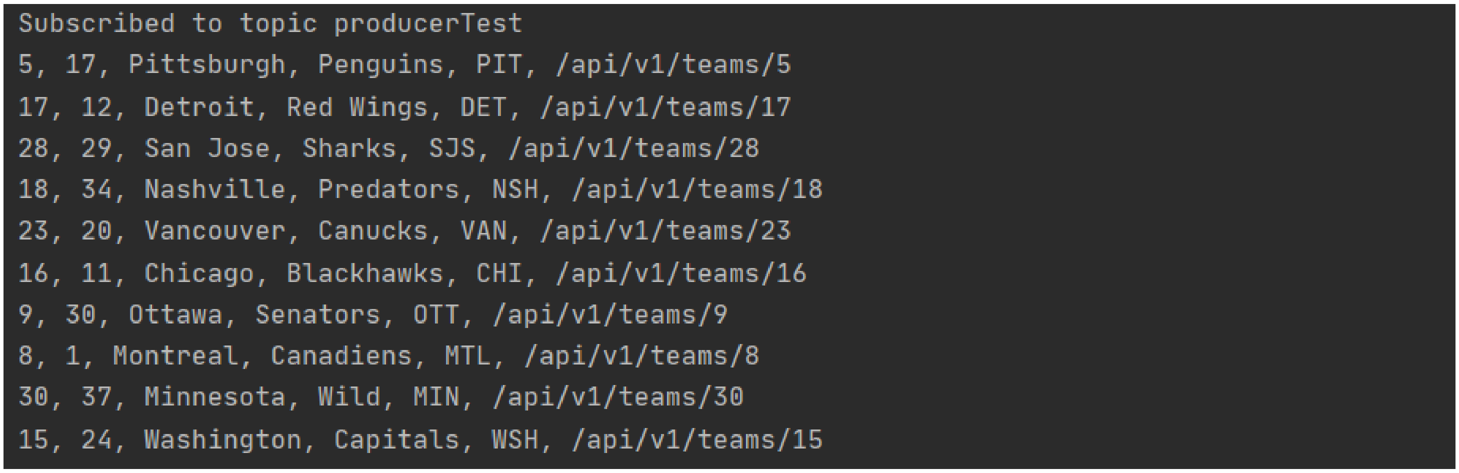
# user can decide the amount of data he needs to publish



```
Enter the line number from which you want to publish data:
5

Enter the line number to which you want to publish data:
15

Process finished with exit code 0
```

Java Output-IntelliJ (Consumer):

# published data will be updated in every 1000ms



```
Subscribed to topic producerTest
5, 17, Pittsburgh, Penguins, PIT, /api/v1/teams/5
17, 12, Detroit, Red Wings, DET, /api/v1/teams/17
28, 29, San Jose, Sharks, SJS, /api/v1/teams/28
18, 34, Nashville, Predators, NSH, /api/v1/teams/18
23, 20, Vancouver, Canucks, VAN, /api/v1/teams/23
16, 11, Chicago, Blackhawks, CHI, /api/v1/teams/16
9, 30, Ottawa, Senators, OTT, /api/v1/teams/9
8, 1, Montreal, Canadiens, MTL, /api/v1/teams/8
30, 37, Minnesota, Wild, MIN, /api/v1/teams/30
15, 24, Washington, Capitals, WSH, /api/v1/teams/15
```