

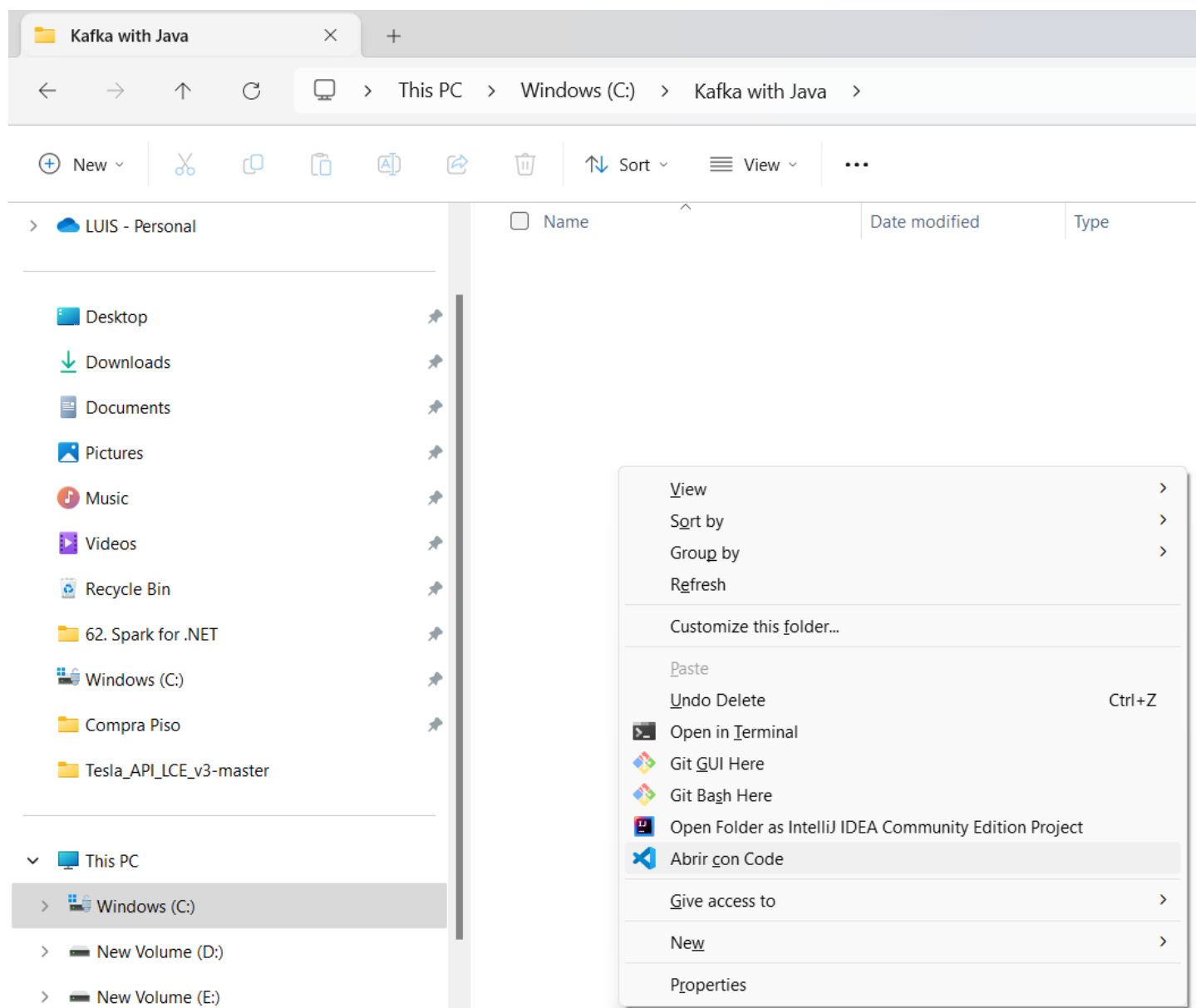
# Kafka Java Consumer application

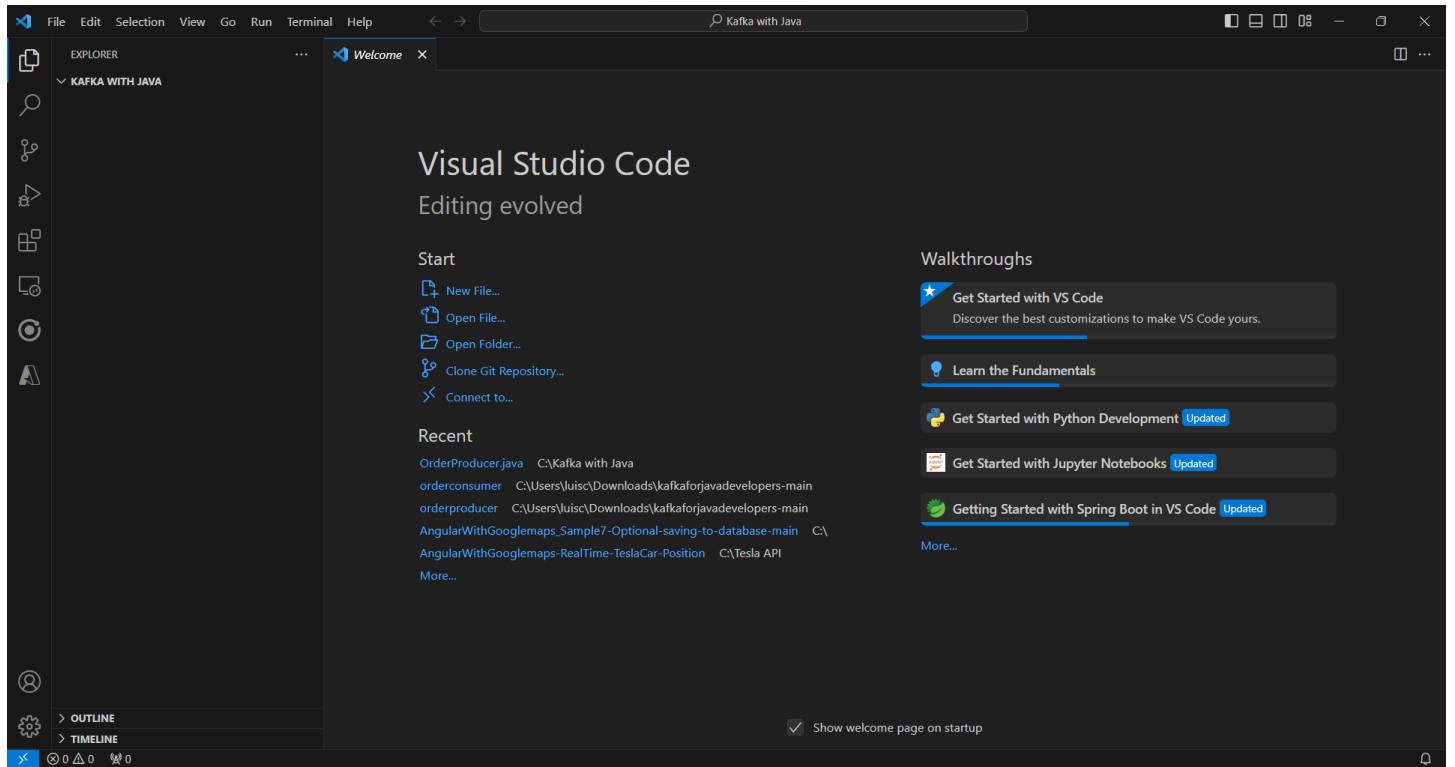
[https://github.com/luiscoco/Kafka\\_Java\\_Consumer](https://github.com/luiscoco/Kafka_Java_Consumer)

## Create a Kafka Consumer Java application with VSCode

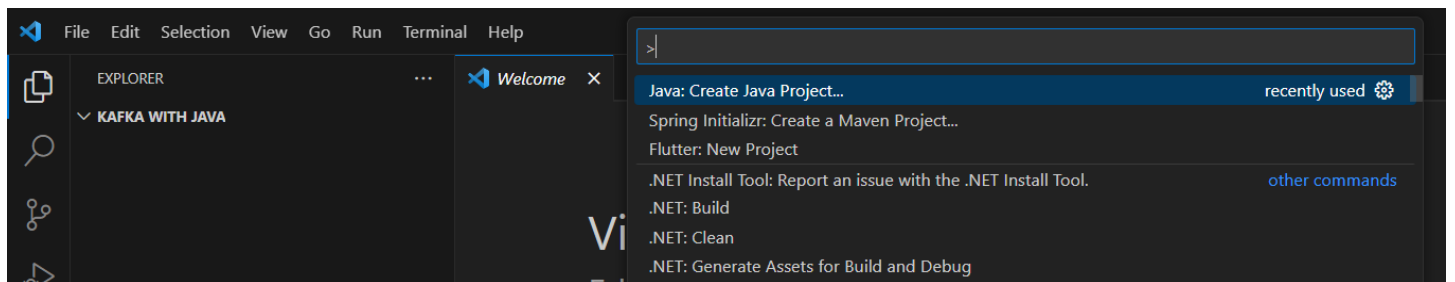
We create a new folder to place the Java application.

We right click inside the folder and we select the option **Open with VSCode**.

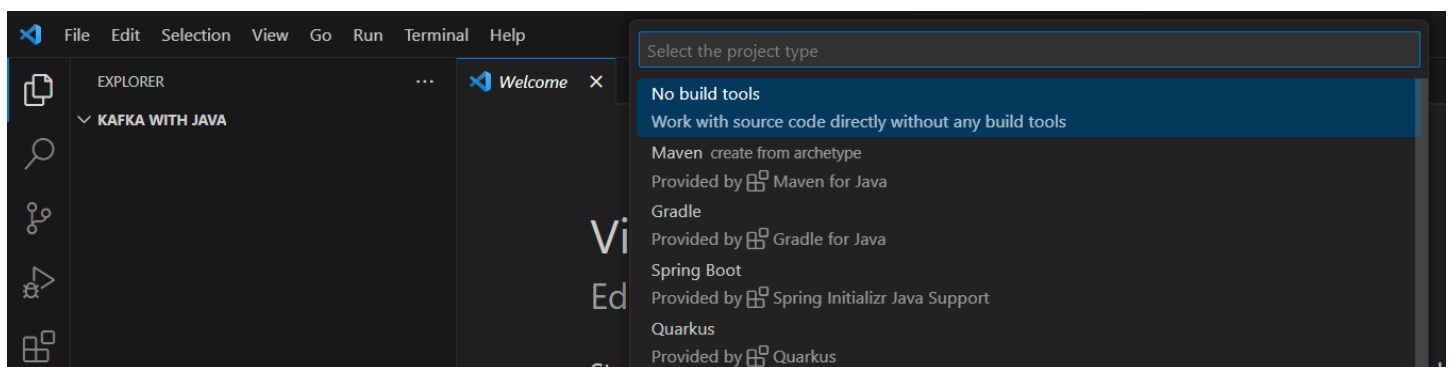




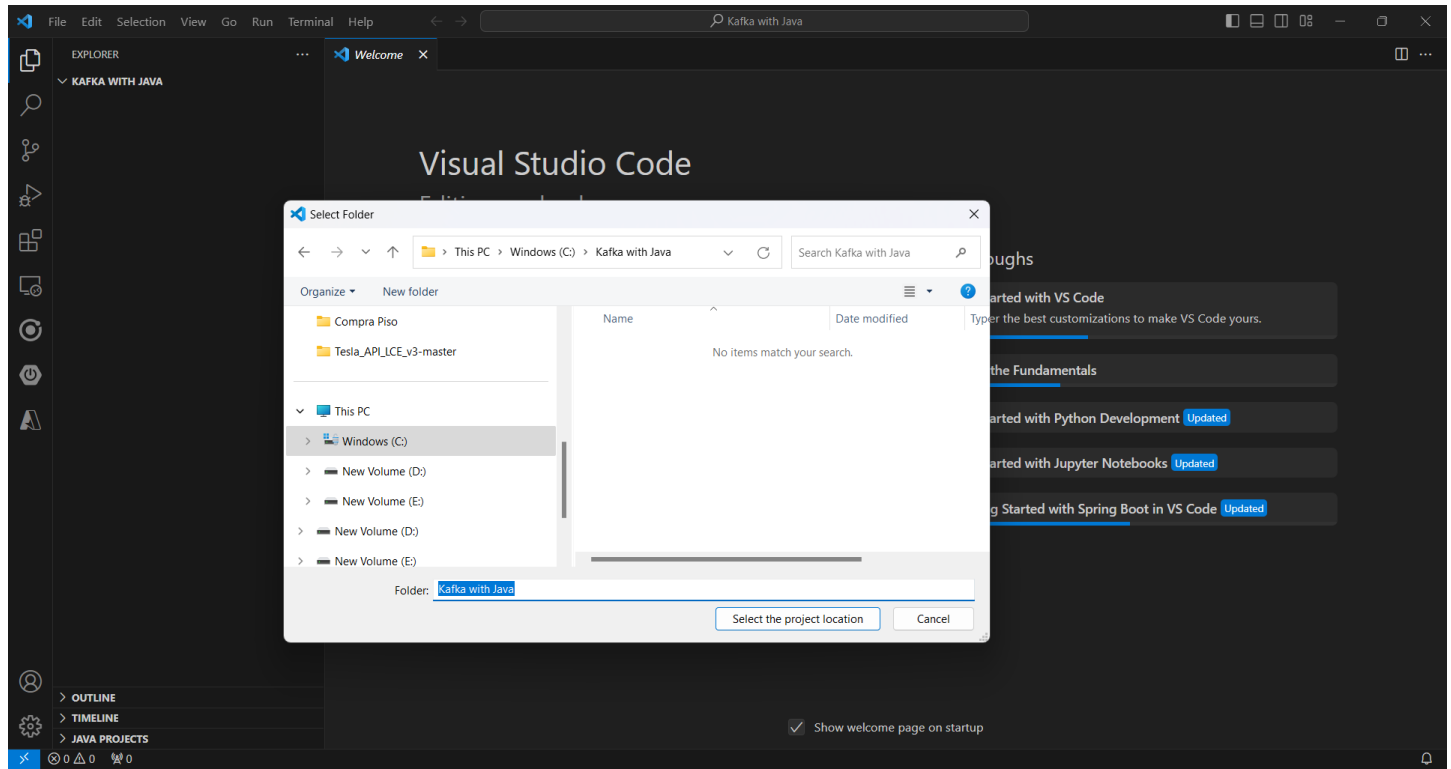
We press the keys **Ctrl+Shift+P** to create a new Java application in VSCode



We select the first option **No build tools**



Now we select the folder where to place the new Java application



Rename the App.java to **KafkaConsumerApp.java**, and then input the following source code:

```
import org.apache.kafka.clients.consumer.Consumer;
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.clients.consumer.ConsumerRecords;
import org.apache.kafka.clients.consumer.KafkaConsumer;
import org.apache.kafka.common.serialization.StringDeserializer;

import java.time.Duration;
import java.util.Collections;
import java.util.Properties;

public class KafkaConsumerApp {
    public static void main(String[] args) {
        // Set up consumer properties
        Properties properties = new Properties();
        properties.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        properties.put(ConsumerConfig.GROUP_ID_CONFIG, "your-group-id");
        properties.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        properties.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);

        // Create Kafka consumer
        Consumer<String, String> consumer = new KafkaConsumer<>(properties);

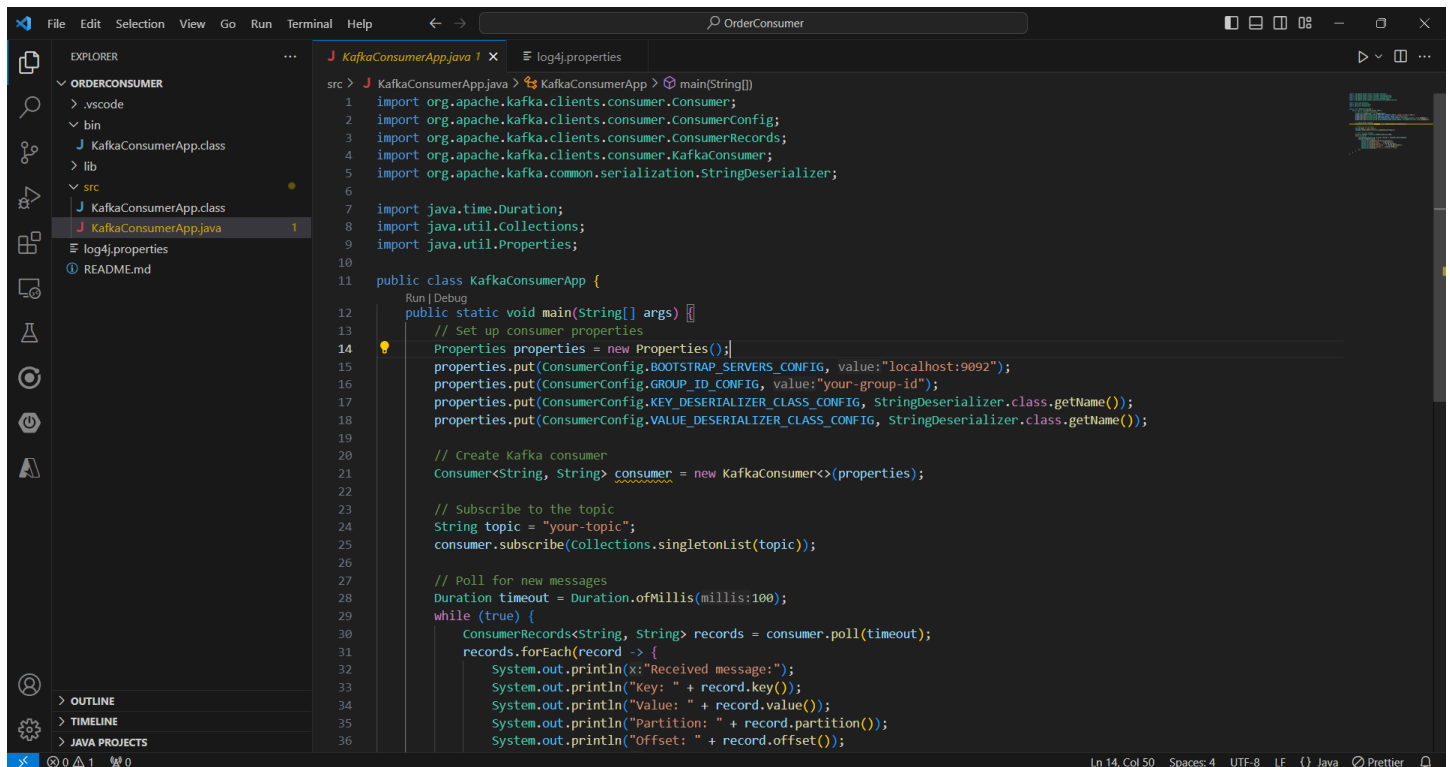
        // Subscribe to the topic
        String topic = "your-topic";
        consumer.subscribe(Collections.singletonList(topic));

        // Poll for new messages
        Duration timeout = Duration.ofMillis(100);
        while (true) {
```

```

ConsumerRecords<String, String> records = consumer.poll(timeout);
records.forEach(record -> {
    System.out.println("Received message:");
    System.out.println("Key: " + record.key());
    System.out.println("Value: " + record.value());
    System.out.println("Partition: " + record.partition());
    System.out.println("Offset: " + record.offset());
    System.out.println("-----");
});
}
}
}

```



Then we create the `log4j.properties` file in the application root:

```
log4j.rootLogger=INFO, stdout
```

```
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
```

```
log4j.appender.stdout.Target=System.out
```

```
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
```

```
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n
```

The screenshot shows the VS Code editor interface. The Explorer panel on the left shows the project structure for 'ORDERCONSUMER', including files like .vscode, bin, KafkaConsumerApp.class, lib, src, KafkaConsumerApp.class, KafkaConsumerApp.java, log4j.properties, and README.md. The main editor area displays the 'log4j.properties' file with the following content:

```

1 log4j.rootLogger=INFO, stdout
2
3 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
4 log4j.appender.stdout.Target=System.out
5 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
6 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n
7

```

Then we **download** Kafka JAR files from Apache Kafka web page (<https://kafka.apache.org/downloads>), and we place the JAR files in the Kafka producer Java application **lib** folder

The screenshot shows the Apache Kafka download page. The browser address bar displays <https://kafka.apache.org/downloads>. The page features the Kafka logo, navigation links for 'GET STARTED', 'DOCS', and 'POWERED BY', and a large 'DOWNLOAD' heading. Below the heading, it states that 3.6.0 is the latest release. It provides instructions on how to verify the download and lists the available download options for version 3.6.0, including source and binary downloads for different Scala versions.

## DOWNLOAD

3.6.0 is the latest release. The current stable version is 3.6.0

You can verify your download by following these [procedures](#) and using these [KEYS](#).

### 3.6.0

- Released Oct 10, 2023
- [Release Notes](#)
- Source download: [kafka-3.6.0-src.tgz](#) ([asc](#), [sha512](#))
- Binary downloads:
  - Scala 2.12 - [kafka\\_2.12-3.6.0.tgz](#) ([asc](#), [sha512](#))
  - Scala 2.13 - [kafka\\_2.13-3.6.0.tgz](#) ([asc](#), [sha512](#))

lib

< > ↑ ↺ > This PC > Windows (C:) > Kafka with Java > OrderProducer > lib

New ✂ Copy Paste Print Share Delete Sort View ...

Home  
Gallery  
LUIS - Personal

Desktop  
Downloads  
Documents  
Pictures  
Music  
Videos  
Recycle Bin  
62. Spark for .NET  
Windows (C:)  
Compra Piso  
Tesla\_API\_LCE\_v3-master

This PC  
Windows (C:)  
New Volume (D:)  
New Volume (E:)

Name	Date modified	Type	Size
activation-1.1.1	29/04/2023 15:35	Executable Jar File	68 KB
aopalliance-repackaged-2.6.1	29/04/2023 15:35	Executable Jar File	27 KB
argparse4j-0.7.0	29/04/2023 15:34	Executable Jar File	89 KB
audience-annotations-0.12.0	02/08/2023 5:07	Executable Jar File	21 KB
caffeine-2.9.3	23/06/2023 7:05	Executable Jar File	891 KB
checker-qual-3.19.0	23/06/2023 7:05	Executable Jar File	217 KB
commons-beanutils-1.9.4	29/04/2023 15:16	Executable Jar File	242 KB
commons-cli-1.4	29/04/2023 15:36	Executable Jar File	53 KB
commons-collections-3.2.2	29/04/2023 15:16	Executable Jar File	575 KB
commons-digester-2.1	29/04/2023 15:16	Executable Jar File	193 KB
commons-io-2.11.0	29/04/2023 15:16	Executable Jar File	320 KB
commons-lang3-3.8.1	29/04/2023 15:35	Executable Jar File	491 KB
commons-logging-1.2	29/04/2023 15:16	Executable Jar File	61 KB
commons-validator-1.7	29/04/2023 15:16	Executable Jar File	186 KB
connect-api-3.6.0	29/09/2023 6:57	Executable Jar File	103 KB
connect-basic-auth-extension-3.6.0	29/09/2023 6:57	Executable Jar File	20 KB
connect-file-3.6.0	29/09/2023 6:57	Executable Jar File	18 KB
connect-json-3.6.0	29/09/2023 6:57	Executable Jar File	34 KB
connect-mirror-3.6.0	29/09/2023 7:00	Executable Jar File	127 KB

lib

< > ↑ ↺

> This PC > Windows (C:) > Kafka with Java > OrderProducer > lib

+ New ✂ 📄 📁 🗑️ ↕ Sort ⌵ View ⌵ ...

> LUIS - Personal

Desktop Downloads Documents Pictures Music Videos Recycle Bin 62. Spark for .NET Windows (C:) Compra Piso Tesla\_API\_LCE\_v3-master

✓ This PC

> Windows (C:)

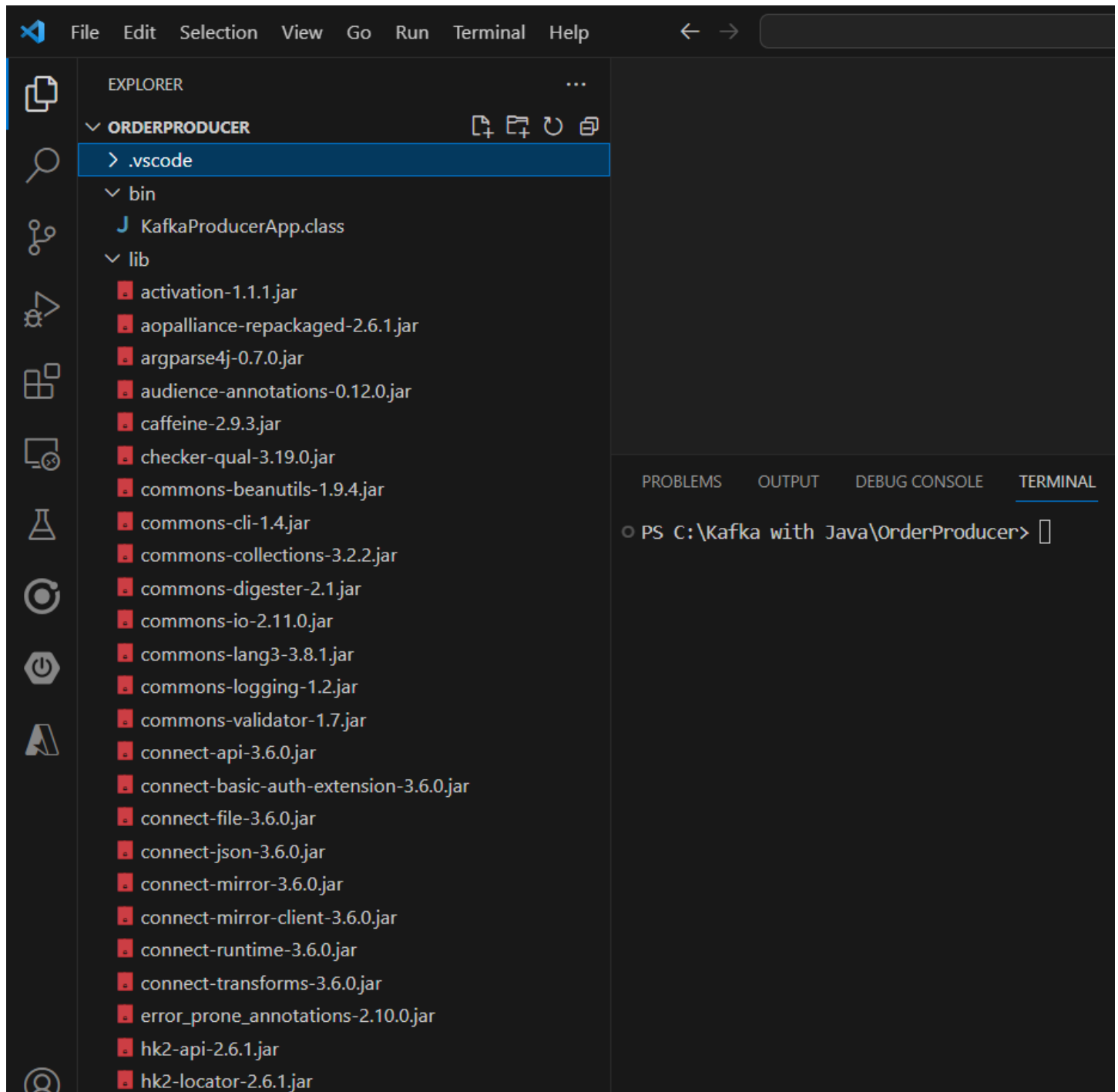
> New Volume (D:)

> New Volume (E:)

> New Volume (D:)

> New Volume (E:)

Name	Date modified	Type	Size
activation-1.1.1	29/04/2023 15:35	Executable Jar File	68 KB
aopalliance-repackaged-2.6.1	29/04/2023 15:35	Executable Jar File	27 KB
argparse4j-0.7.0	29/04/2023 15:34	Executable Jar File	89 KB
audience-annotations-0.12.0	02/08/2023 5:07	Executable Jar File	21 KB
caffeine-2.9.3	23/06/2023 7:05	Executable Jar File	891 KB
checker-qual-3.19.0	23/06/2023 7:05	Executable Jar File	217 KB
commons-beanutils-1.9.4	29/04/2023 15:16	Executable Jar File	242 KB
commons-cli-1.4	29/04/2023 15:36	Executable Jar File	53 KB
commons-collections-3.2.2	29/04/2023 15:16	Executable Jar File	575 KB
commons-digester-2.1	29/04/2023 15:16	Executable Jar File	193 KB
commons-io-2.11.0	29/04/2023 15:16	Executable Jar File	320 KB
commons-lang3-3.8.1	29/04/2023 15:35	Executable Jar File	491 KB
commons-logging-1.2	29/04/2023 15:16	Executable Jar File	61 KB
commons-validator-1.7	29/04/2023 15:16	Executable Jar File	186 KB
connect-api-3.6.0	29/09/2023 6:57	Executable Jar File	103 KB
connect-basic-auth-extension-3.6.0	29/09/2023 6:57	Executable Jar File	20 KB
connect-file-3.6.0	29/09/2023 6:57	Executable Jar File	18 KB
connect-json-3.6.0	29/09/2023 6:57	Executable Jar File	34 KB
connect-mirror-3.6.0	29/09/2023 7:00	Executable Jar File	127 KB



To compile the Kafka Consumer Java application

```
C:\Kafka with Java\OrderConsumer> javac -cp "lib/*;src" src/KafkaConsumerApp.java
```

To run the Kafka Consumer Java application

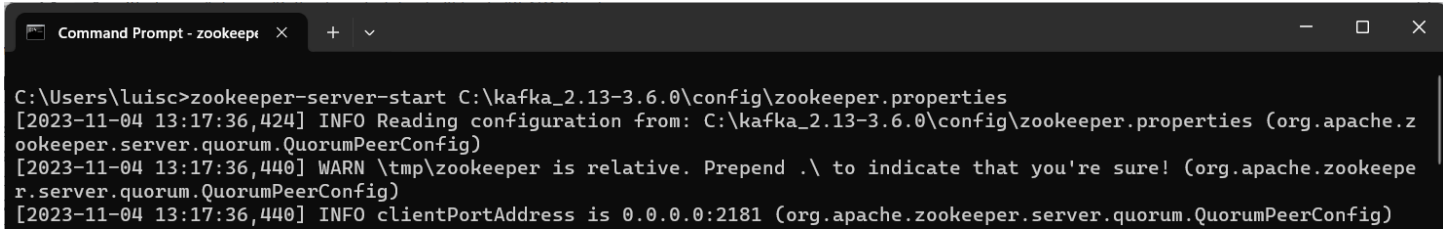
```
C:\Kafka with Java\OrderConsumer> java -cp "lib/*;src;." KafkaConsumerApp
```

## How to run the Consumer and the Producer Java applications

### 1. First we have to run the zookeeper-server-start command



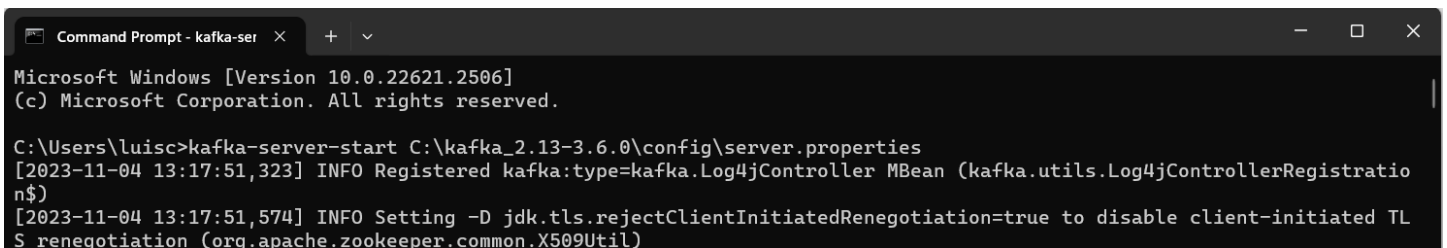
```
zookeeper-server-start C:\kafka_2.13-3.6.0\config\zookeeper.properties
```



```
Command Prompt - zookeep
C:\Users\luisc>zookeeper-server-start C:\kafka_2.13-3.6.0\config\zookeeper.properties
[2023-11-04 13:17:36,424] INFO Reading configuration from: C:\kafka_2.13-3.6.0\config\zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-11-04 13:17:36,440] WARN \tmp\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-11-04 13:17:36,440] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
```

## 2. Open a command prompt window and run the command kafka-server-start

```
kafka-server-start C:\kafka_2.13-3.6.0\config\server.properties
```



```
Command Prompt - kafka-ser
Microsoft Windows [Version 10.0.22621.2506]
(c) Microsoft Corporation. All rights reserved.

C:\Users\luisc>kafka-server-start C:\kafka_2.13-3.6.0\config\server.properties
[2023-11-04 13:17:51,323] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2023-11-04 13:17:51,574] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
```

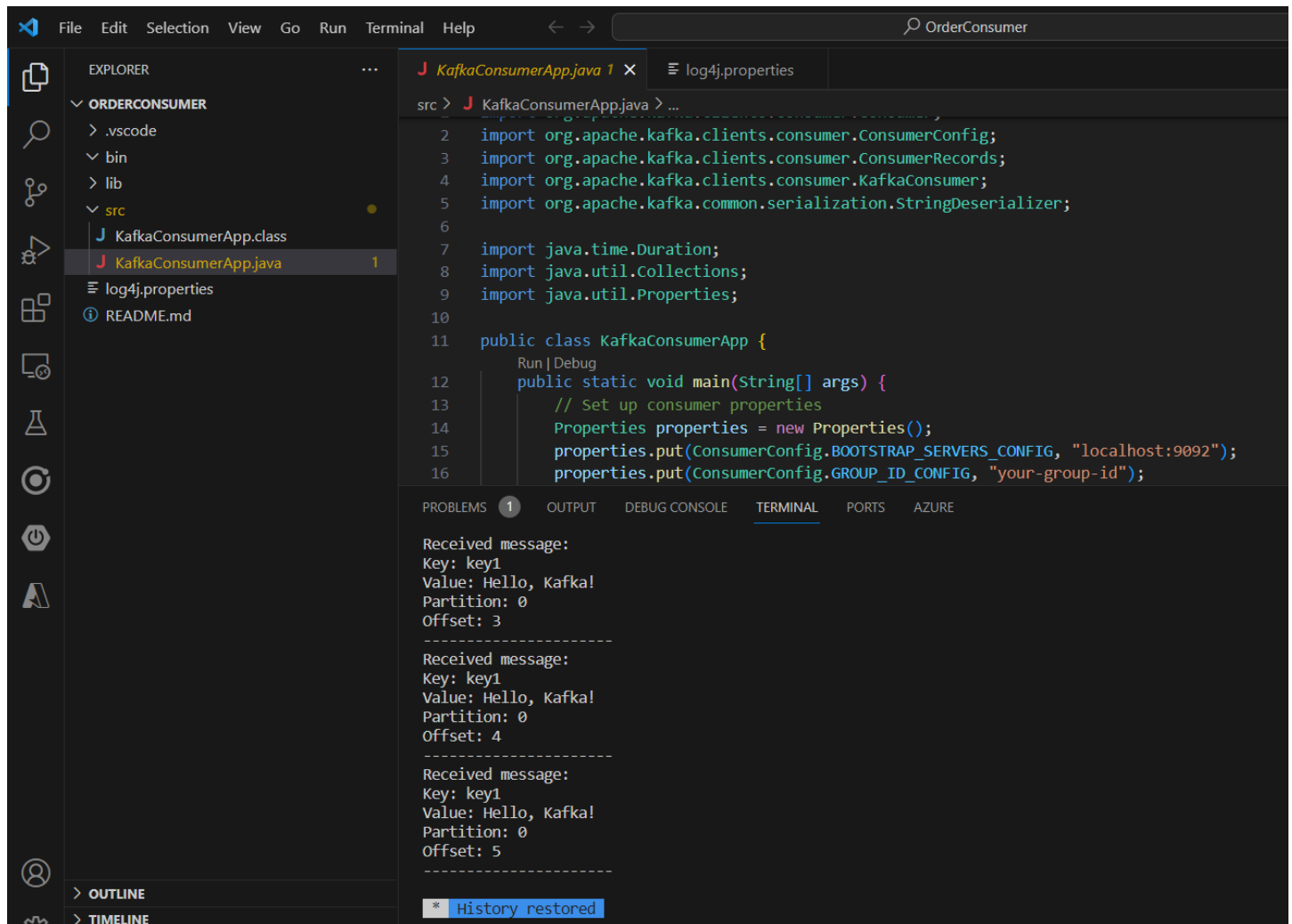
## 3. Then open with VSCode the Kafka Consumer application and run it

To compile the Kafka Consumer Java application

```
C:\Kafka with Java\OrderConsumer> javac -cp "lib/*;src" src/KafkaConsumerApp.java
```

To run the Kafka Consumer Java application

```
C:\Kafka with Java\OrderConsumer> java -cp "lib/*;src;." KafkaConsumerApp
```



#### 4. Then open with VSCode the Kafka Producer application and run it

To compile the Kafka Consumer Java application

```
C:\Kafka with Java\OrderProducer> javac -cp "lib/*;src" src/KafkaProducerApp.java
```

To run the Kafka Producer Java application

```
C:\Kafka with Java\OrderProducer> java -cp "lib/*;src;." KafkaProducerApp
```

The screenshot shows the Visual Studio Code (VS Code) editor interface. The Explorer panel on the left displays the project structure for 'ORDERPRODUCER', including files like .vscode, bin, KafkaProducerApp.class, lib, src, KafkaProducerApp.class, KafkaProducerApp.java, log4j.properties, and README.md. The main editor window shows the source code for 'KafkaProducerApp.java'. The code imports necessary Kafka and Java utilities, defines a 'KafkaProducerApp' class, and implements a 'main' method. The 'main' method sets up producer properties (bootstrap.servers, key.serializer, value.serializer), creates a 'KafkaProducer' instance, produces a message with topic 'your-topic', key 'key1', and value 'Hello Luis From Kafka!', and then sends the message and closes the producer. The bottom panel shows the 'TERMINAL' output, which displays the command executed and the resulting producer configuration values.

```
src > J KafkaProducerApp.java
1 import org.apache.kafka.clients.producer.KafkaProducer;
2 import org.apache.kafka.clients.producer.Producer;
3 import org.apache.kafka.clients.producer.ProducerRecord;
4 import java.util.Properties;
5
6 public class KafkaProducerApp {
7     public static void main(String[] args) {
8         // Set up producer properties
9         Properties properties = new Properties();
10        properties.put(key:"bootstrap.servers", value:"localhost:9092");
11        properties.put(key:"key.serializer", value:"org.apache.kafka.common.serialization.StringSerializer");
12        properties.put(key:"value.serializer", value:"org.apache.kafka.common.serialization.StringSerializer");
13
14        // Create Kafka producer
15        Producer<String, String> producer = new KafkaProducer<>(properties);
16
17        // Produce a message
18        String topic = "your-topic";
19        String key = "key1";
20        String value = "Hello Luis From Kafka!";
21        ProducerRecord<String, String> record = new ProducerRecord<>(topic, key, value);
22
23        // Send the message
24        producer.send(record);
25
26        // Close the producer
27        producer.close();
28    }
29 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

PS C:\Kafka with Java\OrderProducer> java -cp "lib/\*;src;." KafkaProducerApp  
2023-11-04 13:25:18 INFO ProducerConfig:370 - ProducerConfig values:  
acks = -1  
auto.include.jmx.reporter = true  
batch.size = 16384