## **Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**Error! Hyperlink reference not valid.**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.****Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.****Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Message Sender Service (Kafka)**

**Why Kafka?**

Traditional messaging queues like ActiveMQ, RabbitMQ can handle high throughput usually used for long-running or background jobs and communicating between services.

Kafka is a stream-processing platform built by LinkedIn and currently developed under the umbrella of the Apache Software Foundation. Kafka aims to provide low-latency ingestion of large amounts of event data.

**We can use Kafka when we have to move a large amount of data and process it in real-time**. An example would be when we want to process user behavior on our website to generate product suggestions or monitor events produced by our micro-services.

Kafka is built from ground up with horizontal scaling in mind. We can scale by adding more brokers to the existing Kafka cluster.

**Kafka Vocabulary**

Let’s look at the key terminologies of Kafka:

1. **Producer**: A producer is a client that sends messages to the Kafka server to the specified topic.
2. **Consumer**: Consumers are the recipients who receive messages from the Kafka server.
3. **Broker**: Brokers can create a Kafka cluster by sharing information using Zookeeper. A broker receives messages from producers and consumers fetch messages from the broker by topic, partition, and offset.
4. **Cluster**: Kafka is a distributed system. A Kafka cluster contains multiple brokers sharing the workload.
5. **Topic**: A topic is a category name to which messages are published and from which consumers can receive messages.
6. **Partition**: Messages published to a topic are spread across a Kafka cluster into several partitions. Each partition can be associated with a broker to allow consumers to read from a topic in parallel.
7. **Offset**: Offset is a pointer to the last message that Kafka has already sent to a consumer.
8. To work with Kafka, need to install and setup Kafka.
9. To download and install Kafka, please refer the official guide [https://kafka.apache.org/quickstart](https://kafka.apache.org/quickstart" \t "_blank) . Click on the following selected link.



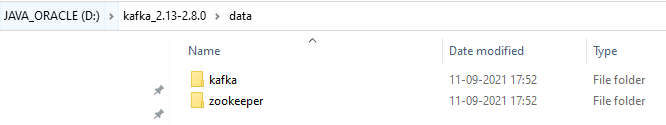
1. Click on the highlighted text to download Kafka: **[kafka\_2.13-2.8.0.tgz](https://dlcdn.apache.org/kafka/2.8.0/kafka_2.13-2.8.0.tgz)**



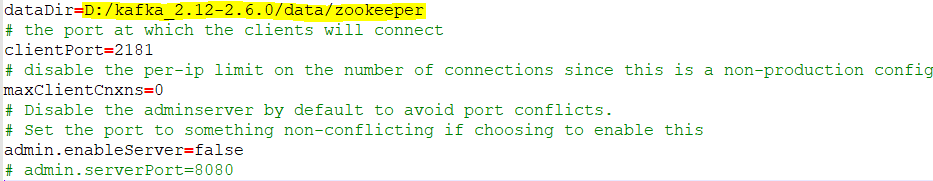
1. Open the powershell and unzip the **[kafka\_2.13-2.8.0.tgz](https://dlcdn.apache.org/kafka/2.8.0/kafka_2.13-2.8.0.tgz)** file.

Command: tar -xf .\kafka\_2.13-2.8.0.tgz

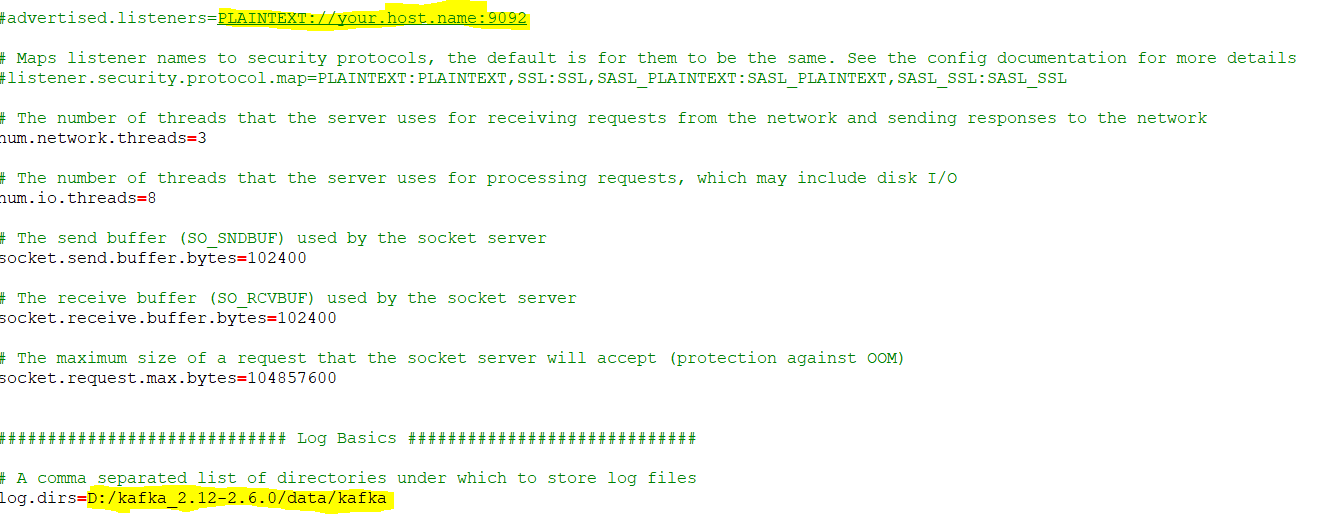
1. **Create data folder and subfolders: zookeeper and kafka in the main directory: kafka\_2.13-2.8.0**



1. **Change the default configuration value**
   1. Update dataDir in zookeeper.properties file located in “D:\kafka\_2.12-2.6.0\config\zookeeper.properties” configuration file.



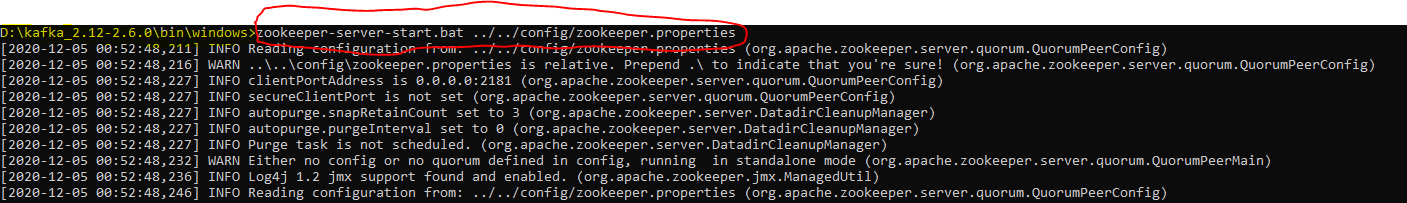
* 1. Update logs.dir in server.properties file located in “D:\kafka\_2.12-2.6.0\config\server.properties” configuration file.



1. **Start Zookeeper**

Now time to start zookeeper from command prompt. Change your directory to **D:\kafka\_2.12-2.6.0\bin\windows** and execute **zookeeper-server-start.bat** command with config/zookeeper.properties configuration file.

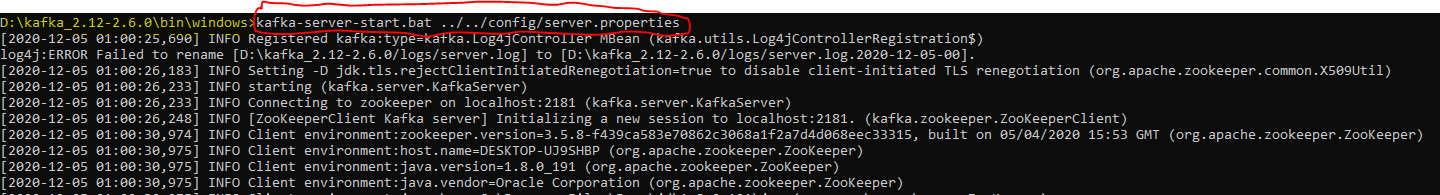
D:\kafka\_2.12-2.6.0\bin\windows>zookeeper-server-start.bat ../../config/zookeeper.properties



1. **Start Kafka**

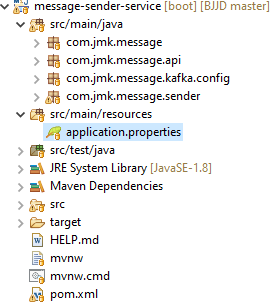
Finally time to start Apache Kafka from command prompt. Change your directory to **D:\kafka\_2.12-2.6.0\bin\windows** and run **kafka-server-start.bat**command with kafka config/server.properties configuration file.

D:\kafka\_2.12-2.6.0\bin\windows>kafka-server-start.bat ../../config/server.properties



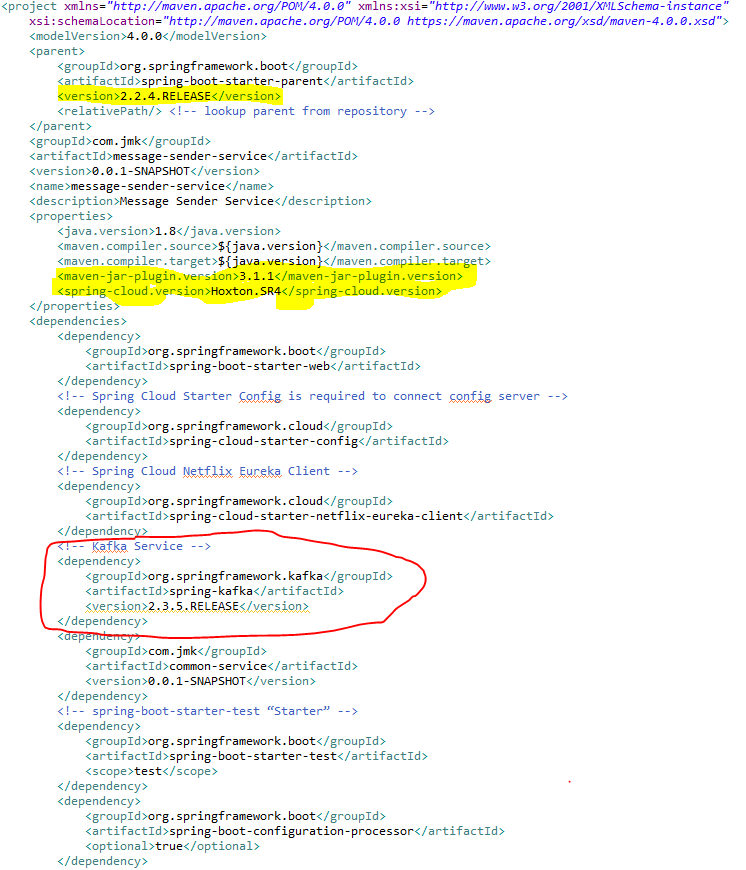
1. Now, by default, the Kafka server starts on localhost:9092.

1. Create the Project using spring Starter Project.
   * Name: kafka-producer-service
   * ArtifactId: kafka-producer-service
   * Package: com.jmk.kafka
2. Click on Next and Finish.



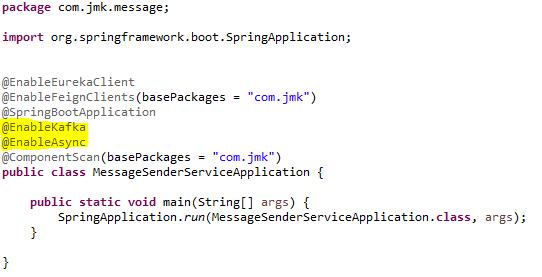
1. Import the Project

* Add the following dependencies in pom.xml with the following considerations:
  + Spring Boot Starter Parent: update the version: 2.2.4.RELEASE highlighted in yellow colour.
  + Spring Cloud version: upgraded to Hoston.SR4. It is highlighted in yellow colour.
  + Maven Jar Plugin Version: After the above updates, pom is giving compiler error in pom.xml so downgraded version of maven jar plugin to 3.1.1. It is highlighted in yellow colour.
  + Spring Kafka:  used to publish and subscribe the messages based on the fault-tolerant messaging system.  It is fast, scalable and distributed by design.





* Update the MessageSenderServiceApplication.java with the following details:
  + Enable Eureka Client so that it can be register with Eureka Server
  + Enable JpaRepositories
  + Enable FeignClient to interact with user-mgmt-service, people-mgmt-service.
  + Enable EnableSwagger2 so that we can view the document api
  + Enable Kafka and Async communication



* application.properties : Make application.properties empty as it is not required here and will be referred in other microservices. Will define the properties of spring.kafka in other microservices.
* Spring Boot does most of the configuration automatically, so we can focus on building the listeners and producing the messages. It also provides the option to override the default configuration through application.properties. The Kafka configuration is controlled by the configuration properties with the prefix spring.kafka.\*:



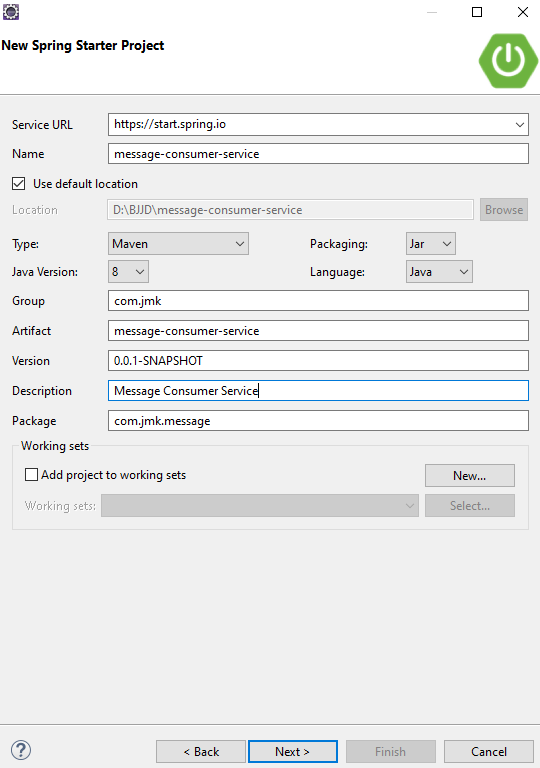
1. To split up responsibilities, we have created two projects:
   * Message-sender-service: It is related to pushing the message on the topic using Spring Producer Configuration. i.e. current section going on.
   * Message-consumer-service: It is related to listening the message received on the topic using kafka consumer configuration. It will be the next topic.
2. Kafka Message Producer Configuration

Let’s have a look at the producer configuration first.

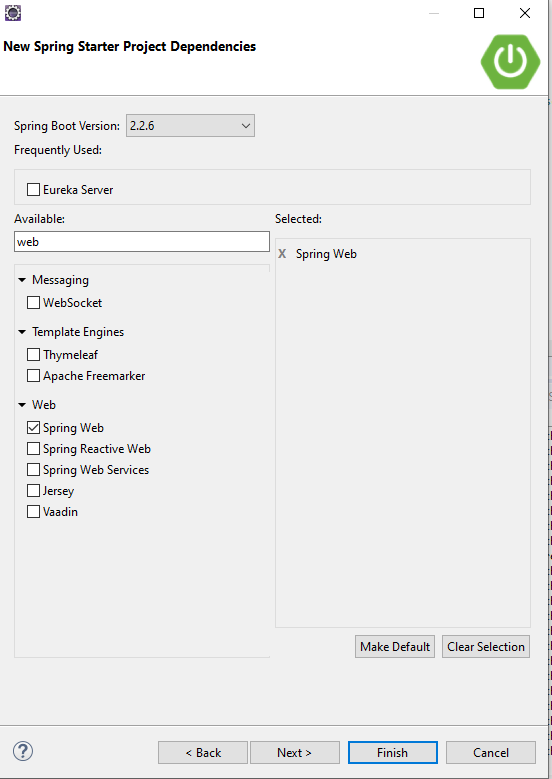
## **Messaging Consumer Service (Kafka and Twilio)**

This service will send the message to the user using the following medium:

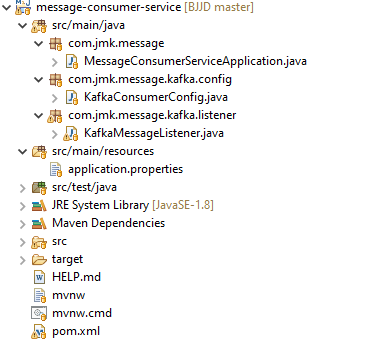
1. WhatsApp
2. Email
3. SMS
4. Create the Project using spring Starter Project.



1. Click on Next and select the Spring Web Module.



1. Click on Finish and import the project.



1. Import the Project

* Add the following dependencies in pom.xml with the following considerations:
  + Spring Boot Starter Parent: update the version: 2.2.4.RELEASE highlighted in yellow colour.
  + Spring Cloud version: upgraded to Hoston.SR4. It is highlighted in yellow colour.
  + Maven Jar Plugin Version: After the above updates, pom is giving compiler error in pom.xml so downgraded version of maven jar plugin to 3.1.1. It is highlighted in yellow colour.
  + Spring Kafka:  used to publish and subscribe the messages based on the fault-tolerant messaging system.  It is fast, scalable and distributed by design.