Day 21 Revisit

* BDD Cucumber & Selenium
* Behaviour Driven Development – Approach to Design Applications
* Integration & e2e Testing
* Testing Pyramid (Unit, Integration, e2e Testing)
* Gherkin – Lang used in BDD (Plain English – Given, When, Then)
* Gherkin Language is used in feature file.
* Feature File, is a combination of many BDD test cases written using Gherkin Lang.
* Scenario – Explains the test.
* Data Table – which provides sample data for testing.
* Cucumber – Is a BDD testing Tool
* Adding Cucumber plugin to Eclipse.
* Selenium Testing Tool (Test Automation Tool)

WebDriver – <https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/119.0.6045.105/win64/chromedriver-win64.zip>

* Selenium is mainly used for Browser based Testing
* Selenium IDE (Integrated Dev Env) – Plugin for Browsers
* Selenium Web Driver – Enables to write the test code in modern prog lang.
* Creating and Executing Cucumber and Selenium Testing.
* Selecting element in selenium (using xpath, using the name, classname, id, style names)

Day 22 Agenda

* Publish/Subscribe Messaging
* Origin Story
* What Is KAFKA
* Why Kafka
* Messages & Batches
* Schema
* Topics & Partitions
* Kafka Client
* Brokers & Clusters
* Multiple Clusters
* Kafka Installation requirements
* Producer/Consumer Pattern (Pub/Sub model)

Pub/sub also called as Producer/Consumer Pattern

Pub/Sub or Publish/Subscribe Messaging Pattern - **publish–subscribe** is a [messaging pattern](https://en.wikipedia.org/wiki/Messaging_pattern) where publishers categorize [messages](https://en.wikipedia.org/wiki/Message_passing) into classes that are received by subscribers.

Message – Data sent from producer to consumers. (Senders & Receivers)

Topics – Every data is associated with a Topic (Category) It maintains all the receivers who are interested in that “Topic”

Subscribers – Receivers (Consumers) of the Data

Publisher – Produces (Creators/Senders) of the data

Kafka - <https://kafka.apache.org/>

<https://kafka.apache.org/downloads>

Event Streaming – Event Source (Databases, log, IoT & Smart devices, Sensors)

Kafka is a Open Source, Distributed Event Streaming platform. (Equal to central nervous system in human body)

Events will have a key, Value, timestamp

Download and Extract kafka from official site

Run the following command

Kafka/bin/windows>zookeeper-server-start.bat ../../config/zookepper.properties

Kafka/bin/windows> kafka-server-start.bat ../../config/server.properties

Kafka.topics.bat --create --topic quickstart-events --bootstrap-server localhost:9092

kafka-console-producer.bat --topic quickstart-events --bootstrap-server localhost:9092

kafka-console-consumer.bat --topic quickstart-events --from-beginning --bootstrap-server localhost:9092

Connect Example

connect-standalone.bat ../../config/connect-standalone.properties ../../config/connect-file-source.properties ../../config/connect-file-sink.properties

Streams Example

kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic streams-plaintext-input

kafka-topics.bat --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic streams-wordcount-output --config cleanup.policy=compact

kafka-topics.bat --bootstrap-server localhost:9092 --describe

kafka-run-class.bat org.apache.kafka.streams.examples.wordcount.WordCountDemo

kafka-console-producer.bat --bootstrap-server localhost:9092 --topic streams-plaintext-input

kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic streams-wordcount-output --from-beginning --formatter kafka.tools.DefaultMessageFormatter --property print.key=true --property print.value=true --property key.deserializer=org.apache.kafka.common.serialization.StringDeserializer --property value.deserializer=org.apache.kafka.common.serialization.LongDeserializer

<https://hilla.dev/blog/full-stack-reactive-kafka-spring-boot-application-tutorial/>

https://howtodoinjava.com/kafka/spring-boot-with-kafka/

<https://www.docker.com/products/docker-desktop/>