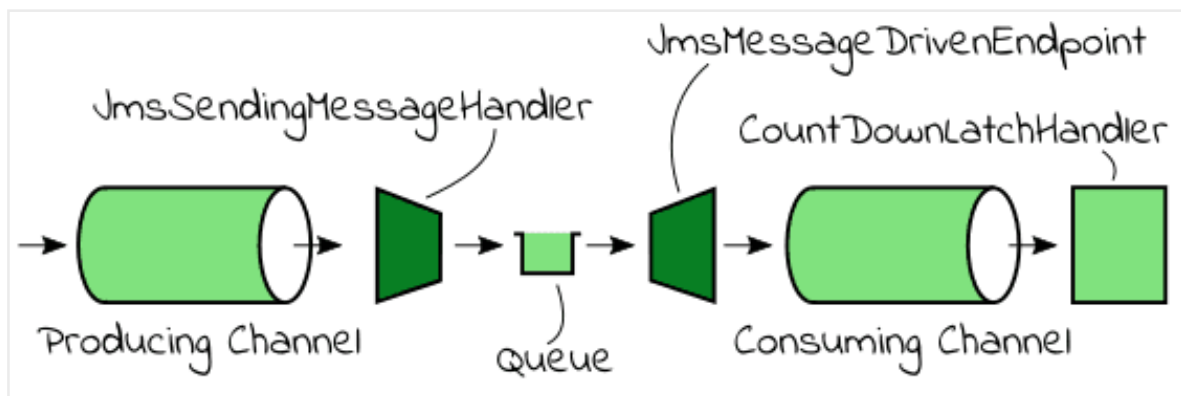


Spring Integration

- Light weight messaging framework
 - JMS (Java Messaging system). MQ, Solace, Kafka
 - Queue and Topic
 - Queue - one to one
 - Topic - one to many
-
- Front office,
 - Middle office
 - Back office



- End points (producer, consumer)
- Channel (Pipe)
- Message

Header

Payload

Different End Points

1. Adapter - Connect ur channel to some other system
2. Filter - Remove some messages from channel based on header and content
3. Transformer - convert a message content or structure
4. Enricher - Add content to the message header or payload
5. Service Activator - Invoke service operation based on the arrival of a message

Message Channel

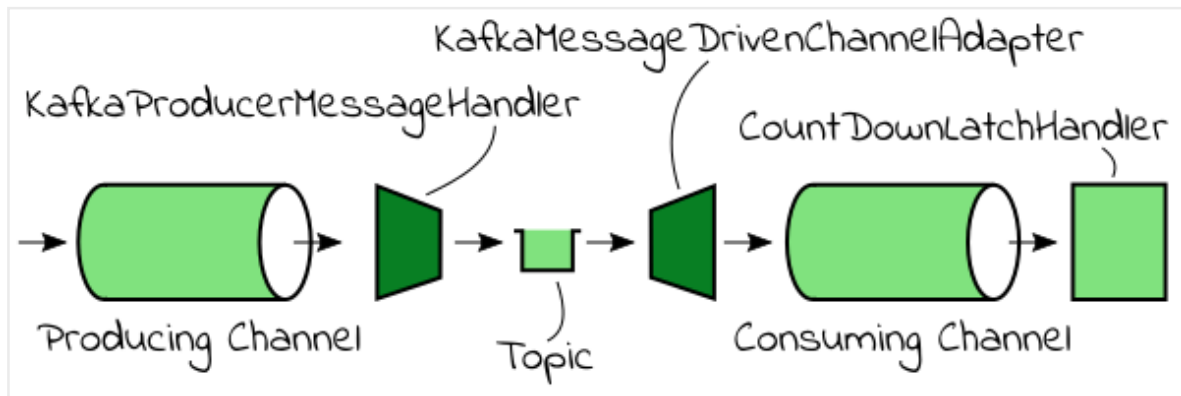
1. Pollable Channel - One to one communication, Consumer actively poll to receive the messages, only one consumer of the channel
2. Subscribe Channel - One to many communication

Adapter

- Connect ur channel to external system
- Provide the bridge between integration framework and external system
- In Built adapters
 - File Adapter
 - JMS Adapter
 - JDBC & JPA Adapter
 - FTP & Secure FTP Adapter
 - Feed Adapter
 - Mail Adapter
 - Mongo DB Adapter
- In Bound Adapter
 - Take messages from external system and get into the channel
- Out Bound Adapter
 - Takes message from channel and deliver it to a external system

Spring Integration with Kafka

- Spring Integration uses the **Message Channel** to pass the information from one component to another. It represents the "pipe" of a [pipes-and-filters architecture](#).
- A Message Channel may follow either [Point-to-Point](#) or [Publish/Subscribe](#) semantics.
- A **Message Endpoint** represents the "filter" of a [pipes-and-filters architecture](#).
- Spring Integration has a number of endpoint types that are supported.
- [Service Activator](#) which simply connects any existing Spring-managed bean to a channel



- Spring Integration Kafka provides a `KafkaProducerMessageHandler` which handles a given message by using a `KafkaTemplate` to send data to Kafka topics.
- By connecting a channel as input to this Message Handler we can send messages to the Kafka bus.
- The second one is a [Channel Adapter](#) endpoint that connects a Message Channel to some other system or transport. Channel Adapters may be either inbound (towards a channel) or outbound (from a channel). Spring Integration Kafka ships with an inbound `KafkaMessageDrivenChannelAdapter` which uses a spring-kafka `KafkaMessageListenerContainer` or `ConcurrentListenerContainer` to receive messages from Kafka topics.
- Our example will consist out of two channels as shown in above diagram. The first *ProducingChannel* will have a `kafkaMessageHandler` that subscribes to the channel and writes all received messages to a 'spring-integration-kafka.t' topic.
- A second *ConsumingChannel* will connect to the same topic using a `KafkaMessageDrivenChannelAdapter`. A custom `CountDownLatchHandler` subscribes to this second channel and lowers a `CountDownLatch` in addition to logging the received message.

Create Topic

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1
--partitions 1 --topic USER_RESOURCE
```

Kafka

```
bin/kafka-server-start.sh config/server.properties
```

Zookeeper

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

```
zookeeper-server-start.bat ..\..\config\zookeeper.properties
```

```
kafka-server-start.bat ..\..\config\server.properties
```

```
kafka-topics.bat --create --topic my-topic --bootstrap-server localhost:9092 --replication-factor 1 --partitions 3
```

```
kafka-console-producer.bat --broker-list localhost:9092 --topic my-topic
```

```
kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic my-topic --from-beginning
```

ZooKeeper Installation

1. Go to your ZooKeeper config directory. C:\kafka\apache-zookeeper-3.7.1-bin\conf
2. Rename file "zoo_sample.cfg" to "zoo.cfg"
3. Open zoo.cfg in any text editor, like Notepad.
4. Find and edit dataDir=/tmp/zookeeper to dataDir=C:\kafka\apache-zookeeper-3.7.1-bin\data
5. Also, add admin.serverPort=9876
6. Add an entry in the System Environment Variables as we did for Java.
 - a. Add ZOOKEEPER_HOME = C:\kafka\apache-zookeeper-3.7.1-bin to the System Variables.
 - b. Edit the System Variable named "Path" and add ;%ZOOKEEPER_HOME%\bin;
7. You can change the default Zookeeper port in zoo.cfg file (Default port 2181).
8. Run ZooKeeper by opening a new cmd and going to C:\kafka\apache-zookeeper-3.7.1-bin\bin and type zkserver.

<https://dzone.com/articles/running-apache-kafka-on-windows-os>

Spring MVC Issue:

1. Update war plugin - 3.3.1
2. Remove the source folder from properties source
3. Update dependency
- 4.