# Kafka Messaging

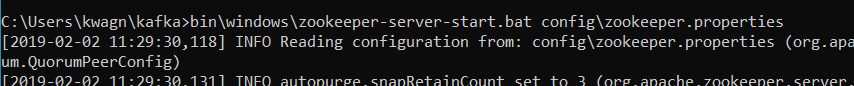
## Kafka Console

### Download & extract

cd C:\Users\kwagn\kafka\bin\windows

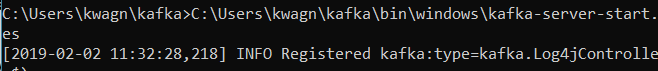
### Start zookeeper

C:\Users\kwagn\kafka\bin\windows\zookeeper-server-start.bat C:\Users\kwagn\kafka\config\zookeeper.properties



### Start kafka

C:\Users\kwagn\kafka\bin\windows\kafka-server-start.bat C:\Users\kwagn\kafka\config\server.properties

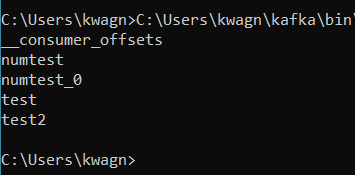


### Create topic

C:\Users\kwagn\kafka\bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test

### Show available topics

C:\Users\kwagn\kafka\bin\windows\kafka-topics.bat --list --zookeeper localhost:2181



### Initiate a producer

C:\Users\kwagn\kafka\bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic test

>input

### Stored in logs

Windows (C:) > tmp > kafka-logs > test-O 
Name 
00000000000000000000. index 
oooooooooooooooooooo.log 
leader-epoch-checkpoint 

### Show the message log

kafka-run-class.bat kafka.tools.DumpLogSegments --deep-iteration --print-data-log --files \tmp\kafka-logs\test-0\00000000000000000000.log

tarting offset: e 
fset: e position: 
0 CreateTime: 1547465113478 isvalid: true keysize: -1 valuesize: 5 magic: 2 compresscodec: 
WNE producerld: 
-1 producerEpoch: -1 sequence: -1 isTransactiona1: false headerKeys: l] payload: Hello 
fset: 1 position: 
73 createTime: 1547465115027 isvalid: true keysize. 
• -1 valuesize: 5 magic: 2 compresscodec 
WNE producerld: 
-1 producerEpoch: 
-1 sequence: -1 isTransactiona1: false headerKeys: [l payload: World 
fset: 2 position: 
146 CreateTime: 154746545624ø isvalid: true keysize: -1 valuesize: 4 magic: 2 compresscode 
WNE producerld: 
-1 producerEpoch: -1 sequence: -1 isTransactiona1: false headerKeys: [l payload: Moin 

### Consuming messages

C:\Users\kwagn\kafka\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test --from-beginning



### Second consumer

C:\Users\kwagn\kafka\bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic test2 --group group2

mand Prompt - 
(c) 2018 Microsoft Corporation. All rights reserved. 
: Wsers : s \kafka - console - 
producer. bat 
- -broker-list --topic test2 
>ha110 
>test 
Command Prompt - - 
:licrosoft Windows [Version 10.0-17134. 523] 
(c) 2e18 Microsoft Corporation. All rights reserved. 
C : \Users \ kwagn>C : \Users \ kwagn -console - consumer. bat 
"ver localhost:gø92 - -topic test2 --from-beginning 
hallo 
test 
(c) 2018 microsoft Corporation. All rights reserved. 
--bQQt 
: : \kafka- console -consumer. bat 
er localhost:9Ø92 - -topic test2 
- -group group2 
- - bootstrap - se 
-bootstrap-serv 

## Kafka Python

<https://towardsdatascience.com/kafka-python-explained-in-10-lines-of-code-800e3e07dad1>

### Start zookeeper, kafka and create topic

C: \Users \Users \kwagn \kafka \bin -topics. bat 
Created topic "numtest" 
- create 
- -zookeeper 2181 
-replication-factor I 
- -partitions I 
-topic numtest 

### Coding a producer / consumer demo

Files are public in github: <https://github.com/kwbln/CS4BD-Edl/tree/master/StreamingKafka>

### Testing

Start zookeeper + start kafka + start demo

#### Broker Problems

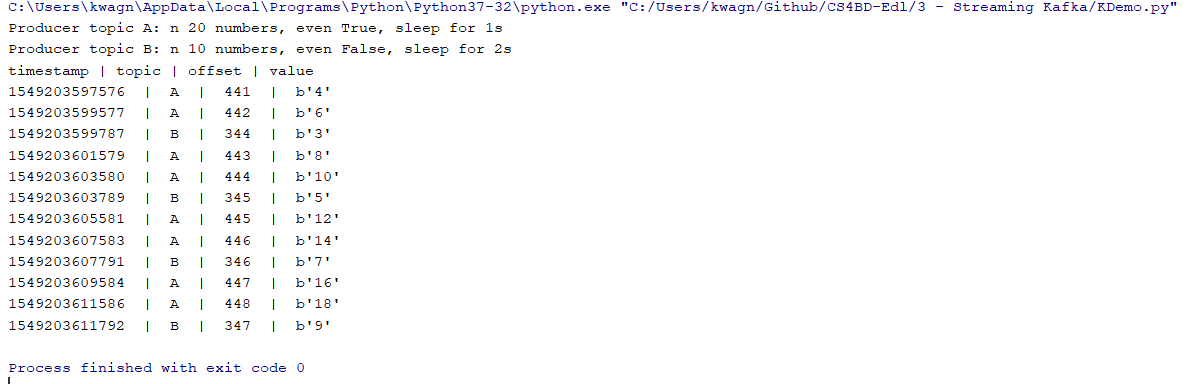
kafka.errors.NoBrokersAvailable: NoBrokersAvailable



<https://stackoverflow.com/questions/47168342/kafka-1-0-stops-with-fatal-shutdown-error-logs-directory-failed>

after restart, the topic need to be recreated

#### Log from a successful run



# Kafka Streams

<https://kafka.apache.org/21/documentation/streams/>

<https://www.confluent.io/blog/introducing-kafka-streams-stream-processing-made-simple/>

<https://jaxenter.de/kafka-0-11-60108>

## Innovation

Kafka offers the possibility to transmit messages from a producer to the broker idempotent. This means that sending operations, which may have to be carried out several times due to problems, result in exactly one written message in the addressed topic. Duplicates can thus be avoided within Topic Partition.

Processing "exactly once" neither leads to a sustained loss of messages nor to unwanted duplicates of messages in topics.

## Previous Kafka versions

The following two semantics were available for processing:

"at most once": In the event that a producer receives no feedback due to a timeout or an error occurs, there is no resending. Affected messages therefore do not end up in the respective topic. In order to be able to exclude potential duplicates by repeated sending, a possible message loss is accepted.

"at least once": Producers can be configured to automatically resend messages in case of timeouts or errors, which can lead to duplicate messages in the topic depending on the error. In order to avoid message loss, possible duplicates are accepted, which may have to be handled later and outside Kafka, depending on the application.

# AlphaGo