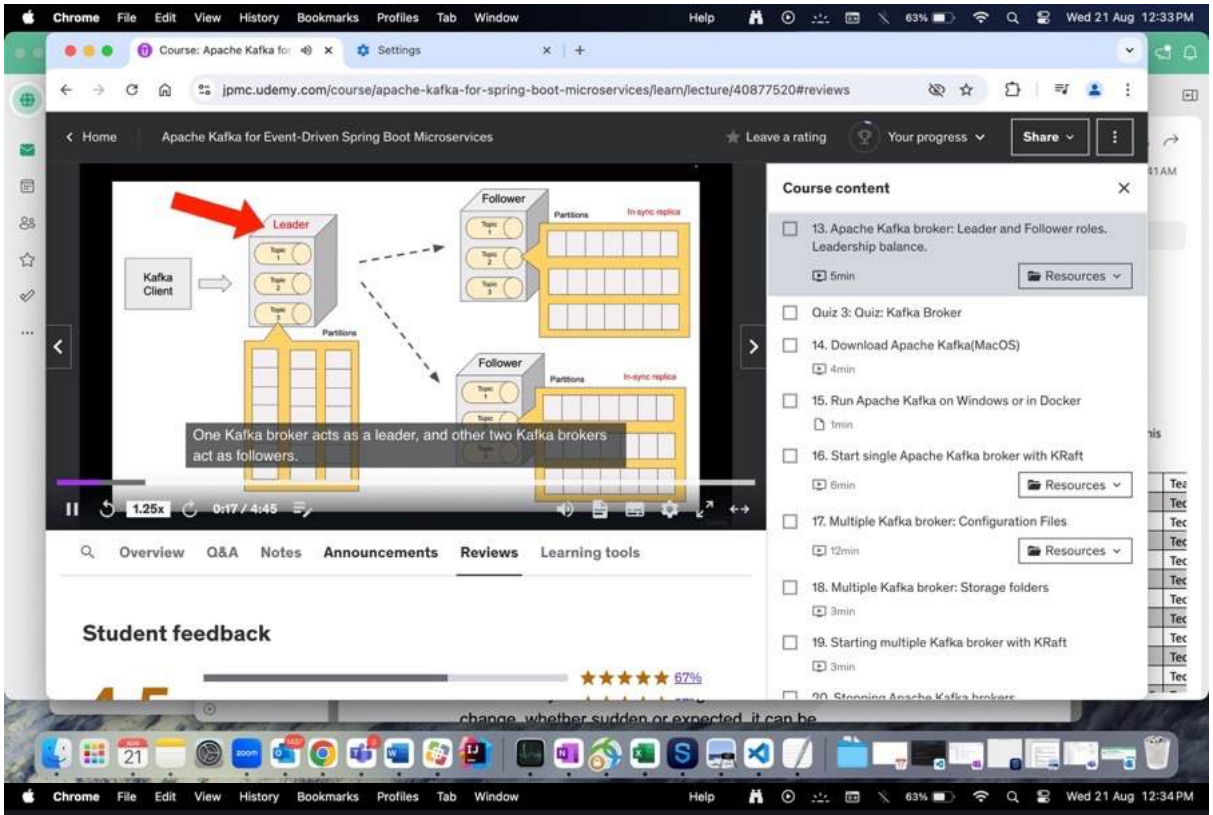
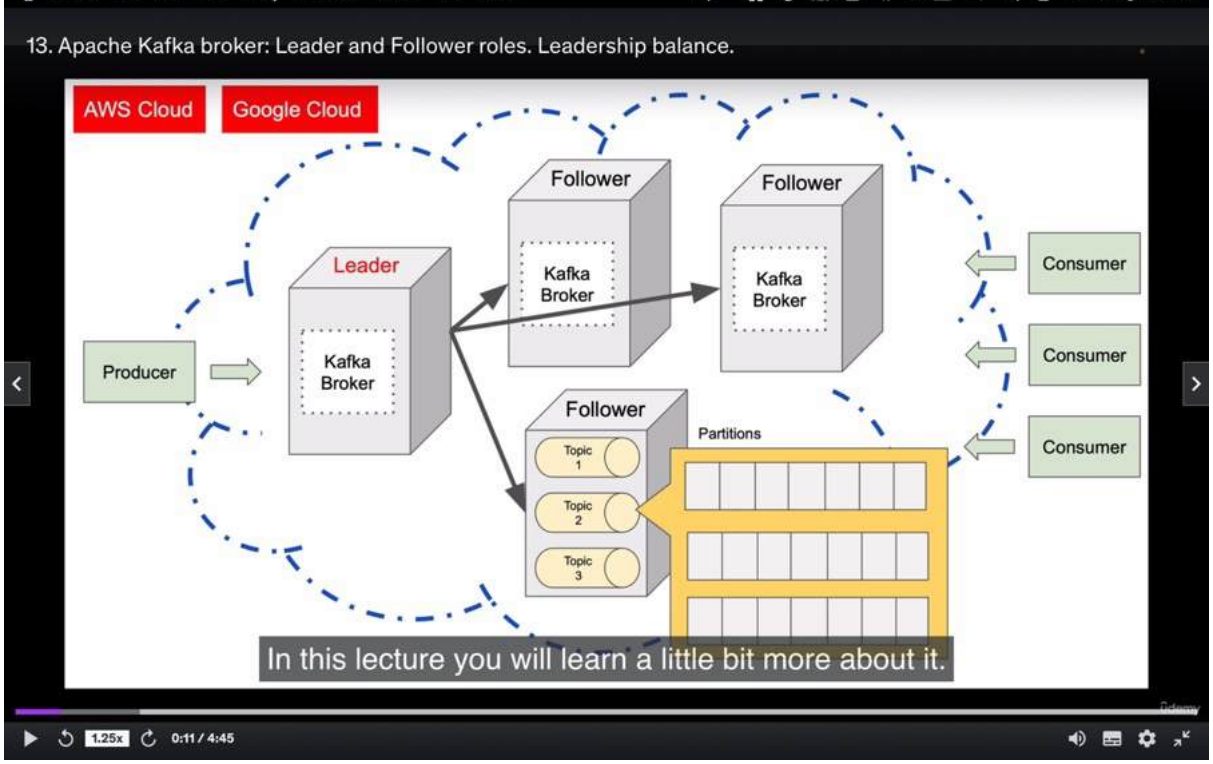
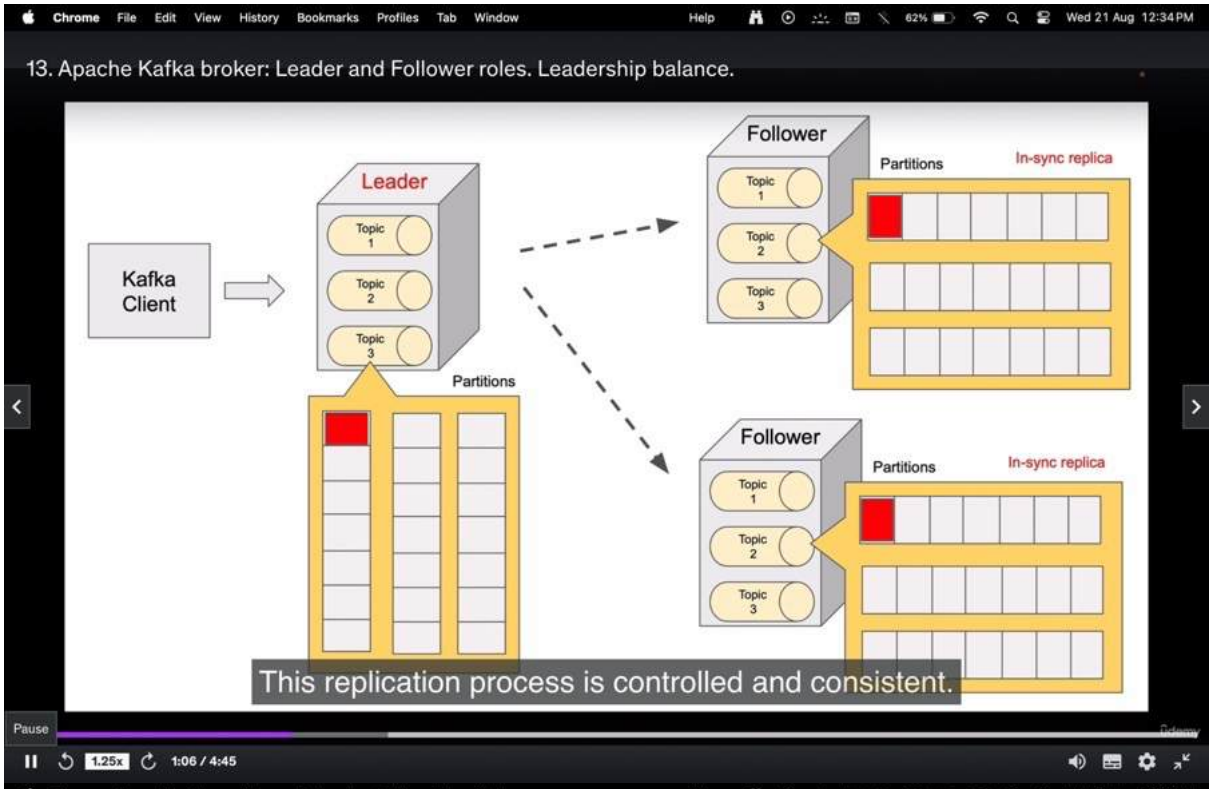
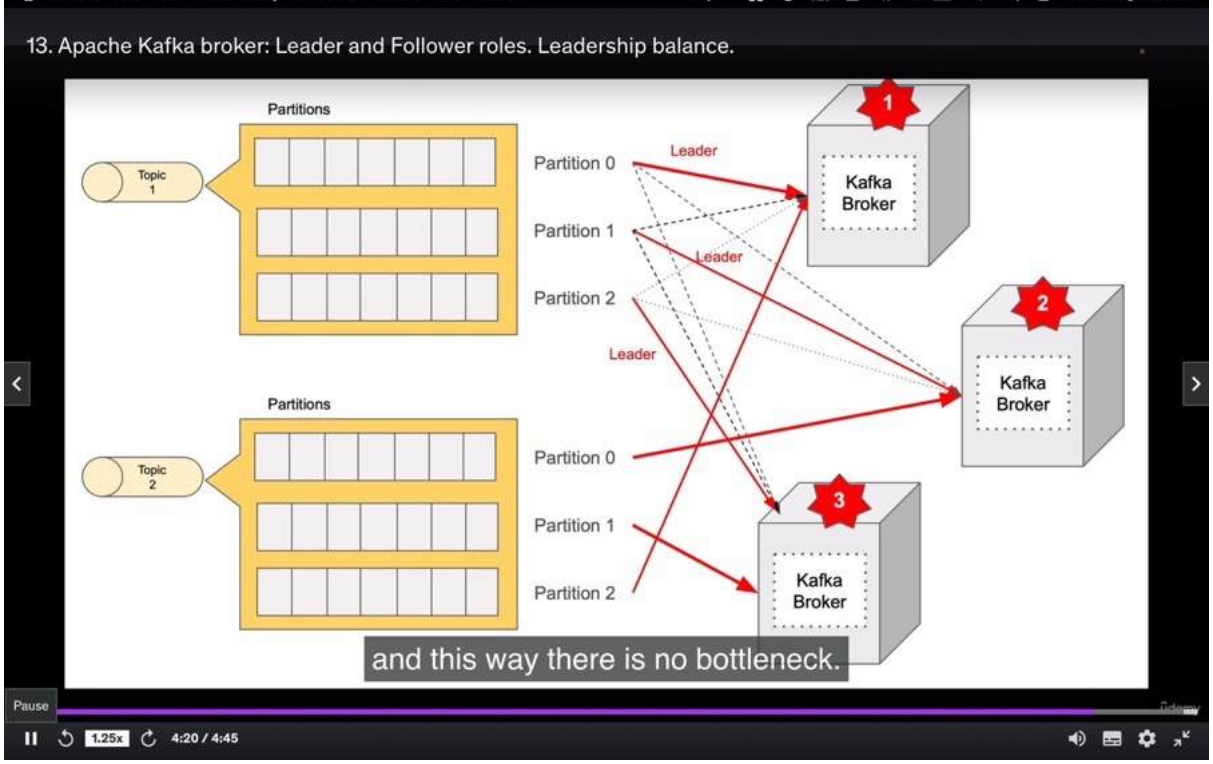

Chrome File Edit View History Bookmarks Profiles Tab Window
Wed 21 Aug 12:33 PM
jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/40877520#reviews
Home
Apache Kafka for Event-Driven Spring Boot Microservices
Leave a rating
Your progress
Share
Kafka Client
Leader
Follower
Partitions
In-sync replica
One Kafka broker acts as a leader, and other two Kafka brokers act as followers.
1.25x
0:17 / 4:45
Course content
13. Apache Kafka broker: Leader and Follower roles. Leadership balance.
5min
Resources
Quiz 3: Quiz: Kafka Broker
14. Download Apache Kafka(MacOS)
4min
15. Run Apache Kafka on Windows or in Docker
1min
16. Start single Apache Kafka broker with KRaft
6min
17. Multiple Kafka broker: Configuration Files
12min
18. Multiple Kafka broker: Storage folders
3min
19. Starting multiple Kafka broker with KRaft
3min
Overview
Q&A
Notes
Announcements
Reviews
Learning tools
Student feedback
67%


Chrome File Edit View History Bookmarks Profiles Tab Window
Wed 21 Aug 12:34 PM
13. Apache Kafka broker: Leader and Follower roles. Leadership balance.
AWS Cloud
Google Cloud
Producer
Leader
Kafka Broker
Follower
Kafka Broker
Follower
Kafka Broker
Follower
Topic 1
Topic 2
Topic 3
Partitions
Consumer
Consumer
Consumer
In this lecture you will learn a little bit more about it.
1.25x
0:11 / 4:45


Chrome File Edit View History Bookmarks Profiles Tab Window Help 62% Wed 21 Aug 12:34 PM
13. Apache Kafka broker: Leader and Follower roles. Leadership balance.
Kafka Client
Leader
Topic 1
Topic 2
Topic 3
Partitions
Follower
Partitions
In-sync replica
Follower
Partitions
In-sync replica
This replication process is controlled and consistent.
Pause
1.25x
1:06 / 4:45


Chrome File Edit View History Bookmarks Profiles Tab Window Help 62% Wed 21 Aug 12:37 PM
13. Apache Kafka broker: Leader and Follower roles. Leadership balance.
Partitions
Topic 1
Partition 0
Partition 1
Partition 2
Leader
Leader
Leader
Kafka Broker
Kafka Broker
Kafka Broker
1
2
3
Partitions
Topic 2
Partition 0
Partition 1
Partition 2
and this way there is no bottleneck.
Pause
1.25x
4:20 / 4:45

16. Start single Apache Kafka broker with KRaft

```
kafka_2.13-3.6.0        kafka_2.13-3.6.0.tar
sergeykargopolov@Sergeys-MacBook-Pro Downloads % mv kafka_2.13-3.6.0 kafka
sergeykargopolov@Sergeys-MacBook-Pro Downloads % ls
kafka                   kafka_2.13-3.6.0.tar
sergeykargopolov@Sergeys-MacBook-Pro Downloads % mv kafka /Users/sergeykargopolov
sergeykargopolov@Sergeys-MacBook-Pro Downloads % cd /Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % ls
LICENSE         bin             libs            site-docs
NOTICE          config          licenses
sergeykargopolov@Sergeys-MacBook-Pro kafka % ls
LICENSE         bin             libs            site-docs
NOTICE          config          licenses
sergeykargopolov@Sergeys-MacBook-Pro kafka % ls config
connect-console-sink.properties     consumer.properties
connect-console-source.properties   kraft
connect-distributed.properties      log4j.properties
connect-file-sink.properties        producer.properties
connect-file-source.properties      server.properties
connect-log4j.properties            tools-log4j.properties
connect-mirror-maker.properties     trogdor.conf
connect-standalone.properties       zookeeper.properties
sergeykargopolov@Sergeys-MacBook-Pro kafka % ls config/kraft
broker.properties       controller.properties   server.properties
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

There are three configuration files.

16. Start single Apache Kafka broker with KRaft

```
kafka-leader-election.sh            zookeeper-server-stop.sh
kafka-log-dirs.sh                   zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro kafka % ls bin/windows
connect-distributed.bat             kafka-leader-election.bat
connect-plugin-path.bat             kafka-log-dirs.bat
connect-standalone.bat              kafka-metadata-quorum.bat
kafka-acls.bat                      kafka-mirror-maker.bat
kafka-broker-api-versions.bat       kafka-producer-perf-test.bat
kafka-cluster.bat                   kafka-reassign-partitions.bat
kafka-configs.bat                   kafka-replica-verification.bat
kafka-console-consumer.bat          kafka-run-class.bat
kafka-console-producer.bat          kafka-server-start.bat
kafka-consumer-groups.bat           kafka-server-stop.bat
kafka-consumer-perf-test.bat        kafka-storage.bat
kafka-delegation-tokens.bat         kafka-streams-application-reset.bat
kafka-delete-records.bat            kafka-topics.bat
kafka-dump-log.bat                  kafka-transactions.bat
kafka-e2e-latency.bat               zookeeper-server-start.bat
kafka-features.bat                  zookeeper-server-stop.bat
kafka-get-offsets.bat               zookeeper-shell.bat
kafka-jmx.bat
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh random-uuid
4hxUhnt4SKCCCTC6br5Cgg
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

This will generate a unique identifier that you will need to use to configure your Kafka cluster.

16. Start single Apache Kafka broker with KRaft

```
connect-distributed.bat                kafka-leader-election.bat
connect-plugin-path.bat                kafka-log-dirs.bat
connect-standalone.bat                 kafka-metadata-quorum.bat
kafka-acls.bat                         kafka-mirror-maker.bat
kafka-broker-api-versions.bat          kafka-producer-perf-test.bat
kafka-cluster.bat                      kafka-reassign-partitions.bat
kafka-configs.bat                      kafka-replica-verification.bat
kafka-console-consumer.bat             kafka-run-class.bat
kafka-console-producer.bat             kafka-server-start.bat
kafka-consumer-groups.bat              kafka-server-stop.bat
kafka-consumer-perf-test.bat           kafka-storage.bat
kafka-delegation-tokens.bat            kafka-streams-application-reset.bat
kafka-delete-records.bat               kafka-topics.bat
kafka-dump-log.bat                     kafka-transactions.bat
kafka-e2e-latency.bat                  zookeeper-server-start.bat
kafka-features.bat                     zookeeper-server-stop.bat
kafka-get-offsets.bat                  zookeeper-shell.bat
kafka-jmx.bat
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh random-uuid
4hxUhnt4SKCCCTC6br5Cgg
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh format -t 4hxUhnt4SKCCCTC6br5Cgg -c con
fig/kraft/server.properties
Formatting /tmp/kraft-combined-logs with metadata.version 3.6-IV2.
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

And now we can try running Kafka Server.

1.25x 5:10 / 6:20

16. Start single Apache Kafka broker with KRaft

```
connect-distributed.bat                kafka-leader-election.bat
connect-plugin-path.bat                kafka-log-dirs.bat
connect-standalone.bat                 kafka-metadata-quorum.bat
kafka-acls.bat                         kafka-mirror-maker.bat
kafka-broker-api-versions.bat          kafka-producer-perf-test.bat
kafka-cluster.bat                      kafka-reassign-partitions.bat
kafka-configs.bat                      kafka-replica-verification.bat
kafka-console-consumer.bat             kafka-run-class.bat
kafka-console-producer.bat             kafka-server-start.bat
kafka-consumer-groups.bat              kafka-server-stop.bat
kafka-consumer-perf-test.bat           kafka-storage.bat
kafka-delegation-tokens.bat            kafka-streams-application-reset.bat
kafka-delete-records.bat               kafka-topics.bat
kafka-dump-log.bat                     kafka-transactions.bat
kafka-e2e-latency.bat                  zookeeper-server-start.bat
kafka-features.bat                     zookeeper-server-stop.bat
kafka-get-offsets.bat                  zookeeper-shell.bat
kafka-jmx.bat
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh random-uuid
4hxUhnt4SKCCCTC6br5Cgg
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh format -t 4hxUhnt4SKCCCTC6br5Cgg -c con
fig/kraft/server.properties
Formatting /tmp/kraft-combined-logs with metadata.version 3.6-IV2.
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-server-start.sh config/kraft/server.properties
```

1.25x 5:57 / 6:20

17. Multiple Kafka broker: Configuration Files

server-3.properties

```
#
# This configuration file is intended for use in KRaft mode, where
# Apache ZooKeeper is not present.
#

############################# Server Basics #############################

# The role of this server. Setting this puts us in KRaft mode
process.roles=broker,controller

# The node id associated with this instance's roles
node.id=3

# The connect string for the controller quorum
controller.quorum.voters=1@localhost:9093

########################### Socket Server Settings ###########################

# The address the socket server listens on.
# Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
# If the broker listener is not defined, the default listener will use a host nam
# with PLAINTEXT listener name, and port 9092.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
```

server-2.properties

```
########################### Socket Server Settings ###########################

# The address the socket server listens on.
# Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
# If the broker listener is not defined, the default listener will use a host nam
# with PLAINTEXT listener name, and port 9092.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
#     listeners = PLAINTEXT://your.host.name:9092
listeners=PLAINTEXT://:9094,CONTROLLER://:9095

# Name of listener used for communication between brokers.
inter.broker.listener.name=PLAINTEXT

# Listener name, hostname and port the broker will advertise to clients.
# If not set, it uses the value for "listeners".
advertised.listeners=PLAINTEXT://localhost:9092

# A comma-separated list of the names of the listeners used by the controller.
# If no explicit mapping set in `listener.security.protocol.map`, default will be
# This is required if running in KRaft mode.
controller.listener.names=CONTROLLER

# Maps listener names to security protocols, the default is for them to be the sa
```

On Windows, instead of 'localhost', you might need to use [::1]
For example:
controller.quorum.voters=1@[::1]:9093

```
25
26
27
28
29
30 controller.quorum.voters=1@localhost:9093
31
32 ############################# Socket Server Settings #############################
33
34 # The address the socket server listens on.
35 # Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
36 # If the broker listener is not defined, the default listener will use a host nam
37 # with PLAINTEXT listener name, and port 9092.
38 #   FORMAT:
39 #     listeners = listener_name://host_name:port
40 #   EXAMPLE:
41 #     listeners = PLAINTEXT://your.host.name:9092
42 listeners=PLAINTEXT://:9092,CONTROLLER://:9093
43
44 # Name of listener used for communication between brokers.
45 inter.broker.listener.name=PLAINTEXT
46
47 # Listener name, hostname and port the broker will advertise to clients.
48 # If not set, it uses the value for "listeners".
49 advertised.listeners=PLAINTEXT://localhost:9092
```

17. Multiple Kafka broker: Configuration Files

```
25
26 # The node id associated with this instance's roles
27 node.id=1
28
29 # The connect string for the controller quorum
30 controller.quorum.voters=1@localhost:9093,2@localhost:9095,3@localhost:9097
31
32 ############################# Socket Server Settings #############################
33
34 # The address the socket server listens on.
35 # Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
36 # If the broker listener is not defined, the default listener will use a host nam
37 # with PLAINTEXT listener name, and port 9092.
38 #   FORMAT:
39 #     listeners = listener_name://host_name:port
40 #   EXAMPLE:
41 #     listeners = PLAINTEXT://your.host.name:9092
42 listeners=PLAINTEXT://:9092,CONTROLLER://:9093
43
44 # Name of listener used for communication between brokers.
45 inter.broker.listener.name=PLAINTEXT
46
47 # Listener name, hostname and port the broker will advertise to clients.
48 # If not set, it uses the value for "listeners".
49 advertised.listeners=PLAINTEXT://localhost:9092
```

1.25x 8:36 / 11:42

17. Multiple Kafka broker: Configuration Files

server-3.properties

```
# The role of this server. Setting this puts us in KRaft mode
process.roles=broker,controller

# The node id associated with this instance's roles
node.id=3

# The connect string for the controller quorum
controller.quorum.voters=1@localhost:9093,2@localhost:9095,3@localhost:9097

############################# Socket Server Settings #############################

# The address the socket server listens on.
# Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
# If the broker listener is not defined, the default listener will use a host nam
# with PLAINTEXT listener name, and port 9092.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
#     listeners = PLAINTEXT://your.host.name:9092
listeners=PLAINTEXT://:9096,CONTROLLER://:9097

# Name of listener used for communication between brokers.
inter.broker.listener.name=PLAINTEXT

# Listener name, hostname and port the broker will advertise to clients.
```

1.25x 9:53 / 11:42

17. Multiple Kafka broker: Configuration Files

server-3.properties

```
# Combined nodes (i.e. those with `process.roles=broker,controller`) must list th
# If the broker listener is not defined, the default listener will use a host nam
# with PLAINTEXT listener name, and port 9092.
#   FORMAT:
#     listeners = listener_name://host_name:port
#   EXAMPLE:
#     listeners = PLAINTEXT://your.host.name:9092
listeners=PLAINTEXT://:9096,CONTROLLER://:9097

# Name of listener used for communication between brokers.
inter.broker.listener.name=PLAINTEXT

# Listener name, hostname and port the broker will advertise to clients.
# If not set, it uses the value for "listeners".
advertised.listeners=PLAINTEXT://localhost:9096

# A comma-separated list of the names of the listeners used by the controller.
# If no explicit mapping set in `listener.security.protocol.map`, default will be
# This is required if running in KRaft mode.
controller.listener.names=CONTROLLER

# Maps listener names to security protocols, the default is for them to be the sa
listener.security.protocol.map=CONTROLLER:PLAINTEXT,PLAINTEXT:PLAINTEXT,SSL:SSL,S

# The number of threads that the server uses for receiving requests from the netw
```

1.25x 10:08 / 11:42

server-1.properties

```
socket.send.buffer.bytes=102400

# The receive buffer (SO_RCVBUF) used by the socket server
socket.receive.buffer.bytes=102400

# The maximum size of a request that the socket server will accept (protection ag
socket.request.max.bytes=104857600


############################# Log Basics #############################

# A comma separated list of directories under which to store log files
log.dirs=/tmp/server-1/kraft-combined-logs

# The default number of log partitions per topic. More partitions allow greater
# parallelism for consumption, but this will also result in more files across
# the brokers.
num.partitions=1

# The number of threads per data directory to be used for log recovery at startup
# This value is recommended to be increased for installations with data dirs loca
num.recovery.threads.per.data.dir=1

############################# Internal Topic Settings  #############################
```

kafka — -zsh — 107×24

```
[2023-11-03 13:34:14,938] INFO Metrics scheduler closed (org.apache.kafka.common.metrics.Metrics)
[2023-11-03 13:34:14,938] INFO Closing reporter org.apache.kafka.common.metrics.JmxReporter (org.apache.kaf
ka.common.metrics.Metrics)
[2023-11-03 13:34:14,938] INFO Metrics reporters closed (org.apache.kafka.common.metrics.Metrics)
[2023-11-03 13:34:14,939] INFO App info kafka.server for 1 unregistered (org.apache.kafka.common.utils.AppI
nfoParser)
[2023-11-03 13:34:14,939] INFO App info kafka.server for 1 unregistered (org.apache.kafka.common.utils.AppI
nfoParser)
sergeykargopolov@Sergeys-MacBook-Pro kafka %
[Restored Nov 3, 2023 at 10:13:26 PM]
Last login: Fri Nov  3 22:02:48 on console
Restored session: Fri  3 Nov 2023 21:54:49 EDT
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh random-uuid
R0HVbO86Rg2Ik4ftri1lkg
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh format -t R0HVbO86Rg2Ik4ftri1lkg -c con
fig/kraft/server-1.properties
Formatting /tmp/server-1/kraft-combined-logs with metadata.version 3.6-IV2.
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh format -t R0HVbO86Rg2Ik4ftri1lkg -c con
fig/kraft/server-2.properties
Formatting /tmp/server-2/kraft-combined-logs with metadata.version 3.6-IV2.
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-storage.sh format -t R0HVbO86Rg2Ik4ftri1lkg -c con
fig/kraft/server-3.properties
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help Thu 22 Aug 1:02 PM

Course: Apache Kafka for | Settings | WorkSpace | Access required

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/39846864#reviews

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

Stopping Kafka Server

- **Control + C**(Mac) or CTRL + C(Windows).
- **kafka-server-stop.sh(Mac) or kafka-server-stop.bat** - Graceful shutdown

1.25x 1:59 / 3:59

Overview Q&A Notes Announcements Reviews Learning tools

Student feedback

★★★★★ 67%

Course content

- **Section 15: Apache Kafka Transactions** 0 / 10 | 53min
- **Section 16: Apache Kafka and Database Transactions** 0 / 6 | 33min
- **Section 17: Integration Testing - Kafka Producer** 0 / 11 | 49min
- **Section 18: Integration Testing - Kafka Consumer** 0 / 6 | 29min
- **Section 19: Saga design pattern with Apache Kafka** 0 / 25 | 1hr 58min
- **Section 20: Saga. Part 2. Compensating Transactions.** 0 / 9 | 29min

Chrome File Edit View History Bookmarks Profiles Tab Window Help Thu 22 Aug 2:41 PM

21. Introduction to Kafka Topic CLI

Kafka CLI: kafka-topics.sh

- Create a new topic,
- List,
- Describe,
- Delete,
- Modify.

Example:

```
bin/kafka-topics.sh --create --topic topic1 --partitions 3 --replication-factor 3 --bootstrap-server localhost:9092
```

And let's learn how to use Apache Kafka topics CLI tool.

1.25x 1:28 / 1:30

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fact
or 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic1.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fact
or 3 --bootstrap-server localhost:9092,localhost:9094
```

Now let's quickly create one more topic and to do that, I will simply repeat this command and will

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fact
or 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic1.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic2 --partitions 3 --replication-fact
or 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic2.
sergeykargopolov@Sergeys-MacBook-Pro bin %
```

And in the following lesson you will learn how to list existing topics and how to describe them to see

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic1.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic2 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic2.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --list --bootstrap-server localhost:9092
topic1
topic2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: topic1   TopicId: Rv9W42kzRKewJ-I6Fr4zTw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic1   Partition: 0    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic1   Partition: 1    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic1   Partition: 2    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
Topic: topic2   TopicId: l41UT3SJThORbGZW87Mmaw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic2   Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
sergeykargopolov@Sergeys-MacBook-Pro bin %
```

To get more information, let's continue to the next lesson.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic1.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic2 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic2.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --list --bootstrap-server localhost:9092
topic1
topic2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: topic1   TopicId: Rv9W42kzRKewJ-I6Fr4zTw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic1   Partition: 0    Le                              3,1,2
        Topic: topic1   Partition: 1    Le                              1,2,3
        Topic: topic1   Partition: 2    Le                              2,3,1
Topic: topic2   TopicId: l41UT3SJThORbGZW87Mmaw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic2   Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --delete --topic topic1 --bootstrap-server localhost:909
2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: topic2   TopicId: l41UT3SJThORbGZW87Mmaw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic2   Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
sergeykargopolov@Sergeys-MacBook-Pro bin %
```

delete.topic.enable=true

But if this configuration property was intentionally set to false, then you will not be able to delete

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic1 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic1.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic topic2 --partitions 3 --replication-fac
tor 3 --bootstrap-server localhost:9092,localhost:9094
Created topic topic2.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --list --bootstrap-server localhost:9092
topic1
topic2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: topic1   TopicId: Rv9W42kzRKewJ-I6Fr4zTw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic1   Partition: 0    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic1   Partition: 1    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic1   Partition: 2    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
Topic: topic2   TopicId: l41UT3SJThORbGZW87Mmaw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic2   Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --delete --topic topic1 --bootstrap-server localhost:909
2
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: topic2   TopicId: l41UT3SJThORbGZW87Mmaw PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: topic2   Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
sergeykargopolov@Sergeys-MacBook-Pro bin %
```

All right, so now that you know how to create and delete Kafka topics, let's continue to the next

26. Introduction to Kafka Producer CLI

Kafka CLI: kafka-console-producer.sh

The main use of kafka-console-producer.sh script is to send a message to a particular Kafka topic.

- Send message with a Key,
- Send message without a Key,
- Send multiple messages from a file.

Example:

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my-topic
```

1.25x 1:10 / 1:21

27. Producing Kafka Message without a Key

```
sergeykargopolov@Sergeys-MacBook-Pro ~ % cd kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % cd bin
sergeykargopolov@Sergeys-MacBook-Pro bin % ls
connect-distributed.sh                kafka-metadata-quorum.sh
connect-mirror-maker.sh               kafka-metadata-shell.sh
connect-plugin-path.sh                kafka-mirror-maker.sh
connect-standalone.sh                 kafka-producer-perf-test.sh
kafka-acls.sh                         kafka-reassign-partitions.sh
kafka-broker-api-versions.sh          kafka-replica-verification.sh
kafka-cluster.sh                      kafka-run-class.sh
kafka-configs.sh                      kafka-server-start.sh
kafka-console-consumer.sh             kafka-server-stop.sh
kafka-console-producer.sh             kafka-storage.sh
kafka-consumer-groups.sh              kafka-streams-application-reset.sh
kafka-consumer-perf-test.sh           kafka-topics.sh
kafka-delegation-tokens.sh            kafka-transactions.sh
kafka-delete-records.sh               kafka-verifiable-consumer.sh
kafka-dump-log.sh                     kafka-verifiable-producer.sh
kafka-e2e-latency.sh                  trogdor.sh
kafka-features.sh                     windows
kafka-get-offsets.sh                  zookeeper-security-migration.sh
kafka-jmx.sh                          zookeeper-server-start.sh
kafka-leader-election.sh              zookeeper-server-stop.sh
kafka-log-dirs.sh                     zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
```

If I hit enter to run this command now and try sending messages to this topic, I will get an error

1.25x 1:26 / 4:29

27. Producing Kafka Message without a Key

```
kafka-configs.sh                      kafka-server-start.sh
kafka-console-consumer.sh             kafka-server-stop.sh
kafka-console-producer.sh             kafka-storage.sh
kafka-consumer-groups.sh              kafka-streams-application-reset.sh
kafka-consumer-perf-test.sh           kafka-topics.sh
kafka-delegation-tokens.sh            kafka-transactions.sh
kafka-delete-records.sh               kafka-verifiable-consumer.sh
kafka-dump-log.sh                     kafka-verifiable-producer.sh
kafka-e2e-latency.sh                  trogdor.sh
kafka-features.sh                     windows
kafka-get-offsets.sh                  zookeeper-security-migration.sh
kafka-jmx.sh                          zookeeper-server-start.sh
kafka-leader-election.sh              zookeeper-server-stop.sh
kafka-log-dirs.sh                     zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
```

But even though I got this error messages, I can still continue sending messages.

1.25x 1:45 / 4:29

27. Producing Kafka Message without a Key

```
kafka-console-consumer.sh          kafka-server-stop.sh
kafka-console-producer.sh          kafka-storage.sh
kafka-consumer-groups.sh           kafka-streams-application-reset.sh
kafka-consumer-perf-test.sh        kafka-topics.sh
kafka-delegation-tokens.sh         kafka-transactions.sh
kafka-delete-records.sh            kafka-verifiable-consumer.sh
kafka-dump-log.sh                  kafka-verifiable-producer.sh
kafka-e2e-latency.sh               trogdor.sh
kafka-features.sh                  windows
kafka-get-offsets.sh               zookeeper-security-migration.sh
kafka-jmx.sh                       zookeeper-server-start.sh
kafka-leader-election.sh           zookeeper-server-stop.sh
kafka-log-dirs.sh                  zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PA
>Hello world 2
```

And this is because by default, Kafka is configured to automatically create a new topic if that topic

1.25x 2:01 / 4:29

27. Producing Kafka Message without a Key

```
kafka-console-consumer.sh          kafka-server-stop.sh
kafka-console-producer.sh          kafka-storage.sh
kafka-consumer-groups.sh           kafka-streams-application-reset.sh
kafka-consumer-perf-test.sh        kafka-topics.sh
kafka-delegation-tokens.sh         kafka-transactions.sh
kafka-delete-records.sh            kafka-verifiable-consumer.sh
kafka-dump-log.sh                  kafka-verifiable-producer.sh
kafka-e2e-latency.sh               trogdor.sh
kafka-features.sh                  windows
kafka-get-offsets.sh               zookeeper-security-migration.sh
kafka-jmx.sh                       zookeeper-server-start.sh
kafka-leader-election.sh           zookeeper-server-stop.sh
kafka-log-dirs.sh
sergeykargopolov@Sergeys-MacBook-P                                  erver localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PA
>Hello world 2
```

auto.create.topics.enable=true

This is the default configuration and it is controlled with the following property in the broker configuration.

1.25x 2:21 / 4:29

27. Producing Kafka Message without a Key

```
kafka-delegation-tokens.sh          kafka-transactions.sh
kafka-delete-records.sh             kafka-verifiable-consumer.sh
kafka-dump-log.sh                   kafka-verifiable-producer.sh
kafka-e2e-latency.sh                trogdor.sh
kafka-features.sh                   windows
kafka-get-offsets.sh                zookeeper-security-migration.sh
kafka-jmx.sh                        zookeeper-server-start.sh
kafka-leader-election.sh            zookeeper-server-stop.sh
kafka-log-dirs.sh                   zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
>Hello world 2
>Hello world 3
>^C
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello
```

Hello world!

1.25x 4:02 / 4:29

27. Producing Kafka Message without a Key

```
kafka-console-producer.sh           kafka-storage.sh
kafka-consumer-groups.sh            kafka-streams-application-reset.sh
kafka-consumer-perf-test.sh         kafka-topics.sh
kafka-delegation-tokens.sh          kafka-transactions.sh
kafka-delete-records.sh             kafka-verifiable-consumer.sh
kafka-dump-log.sh                   kafka-verifiable-producer.sh
kafka-e2e-latency.sh                trogdor.sh
kafka-features.sh                   windows
kafka-get-offsets.sh                zookeeper-security-migration.sh
kafka-jmx.sh                        zookeeper-server-start.sh
kafka-leader-election.sh            zookeeper-server-stop.sh
kafka-log-dirs.sh                   zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafk                    --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
>Hello world 2
>Hello world 3
```

Contol + C

You can hold control button on your keyboard and then press C.

1.25x 3:48 / 4:30

27. Producing Kafka Message without a Key

```
kafka-delegation-tokens.sh          kafka-transactions.sh
kafka-delete-records.sh             kafka-verifiable-consumer.sh
kafka-dump-log.sh                   kafka-verifiable-producer.sh
kafka-e2e-latency.sh                trogdor.sh
kafka-features.sh                   windows
kafka-get-offsets.sh                zookeeper-security-migration.sh
kafka-jmx.sh                        zookeeper-server-start.sh
kafka-leader-election.sh            zookeeper-server-stop.sh
kafka-log-dirs.sh                   zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
>Hello world 2
>Hello world 3
>^C
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>
```

Now my script is running again and I can send more messages.

1.25x 3:59 / 4:30

```
kafka-delete-records.sh             kafka-verifiable-consumer.sh
kafka-dump-log.sh                   kafka-verifiable-producer.sh
kafka-e2e-latency.sh                trogdor.sh
kafka-features.sh                   windows
kafka-get-offsets.sh                zookeeper-security-migration.sh
kafka-jmx.sh                        zookeeper-server-start.sh
kafka-leader-election.sh            zookeeper-server-stop.sh
kafka-log-dirs.sh                   zookeeper-shell.sh
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world
[2023-11-08 18:35:02,329] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 5 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,430] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 6 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,535] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 7 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,638] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 8 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,741] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 9 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
[2023-11-08 18:35:02,852] WARN [Producer clientId=console-producer] Error while fetching metadata with correlation id
 10 : {my-topic=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clients.NetworkClient)
>Hello world 2
>Hello world 3
>^C
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic
>Hello world 4
>
```

So if you want to store and read messages in order, you will need to use the same key for those messages.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:9094 --topic my-topic --property "parse.key=true"
```

When the passkey property is set to true, producer expects each message to be in the format of key

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:9094 --topic my-topic --property "parse.key=true" --property "key.separator=:"
```

And this is it.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:9094 --topic my-topic --property "parse.key=true" --property "key.separator=:"
>firstName:Sergey
```

So in this case the string on the left side of a colon is a key, and the string on the right side of

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:9094 --topic my-topic --property "parse.key=true" --property "key.separator=:"
>firstName:Sergey
```

Key

Value

the colon is a value.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092,localhost:90
94 --topic my-topic --property "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>test message
org.apache.kafka.common.KafkaException: No key separator found on line number 2: 'test message'
        at kafka.tools.ConsoleProducer$LineMessageReader.kafka$tools$ConsoleProducer$LineMessageReader$$parse(Console
Producer.scala:437)
        at kafka.tools.ConsoleProducer$LineMessageReader$$anon$3.hasNext(ConsoleProducer.scala:407)
        at kafka.tools.ConsoleProducer$.loopReader(ConsoleProducer.scala:90)
        at kafka.tools.ConsoleProducer$.main(ConsoleProducer.scala:99)
        at kafka.tools.ConsoleProducer.main(ConsoleProducer.scala)
sergeykargopolov@Sergeys-MacBook-Pro bin %
```

And I got an error that says no key separator found on line number two test message all right.

Kafka CLI: kafka-console-consumer.sh

The main use of kafka-console-consumer.sh is to fetch and display messages from a Kafka topic to your terminal.

- Read new messages only,
- Read all messages from the beginning.

Example:

```
bin/kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
```

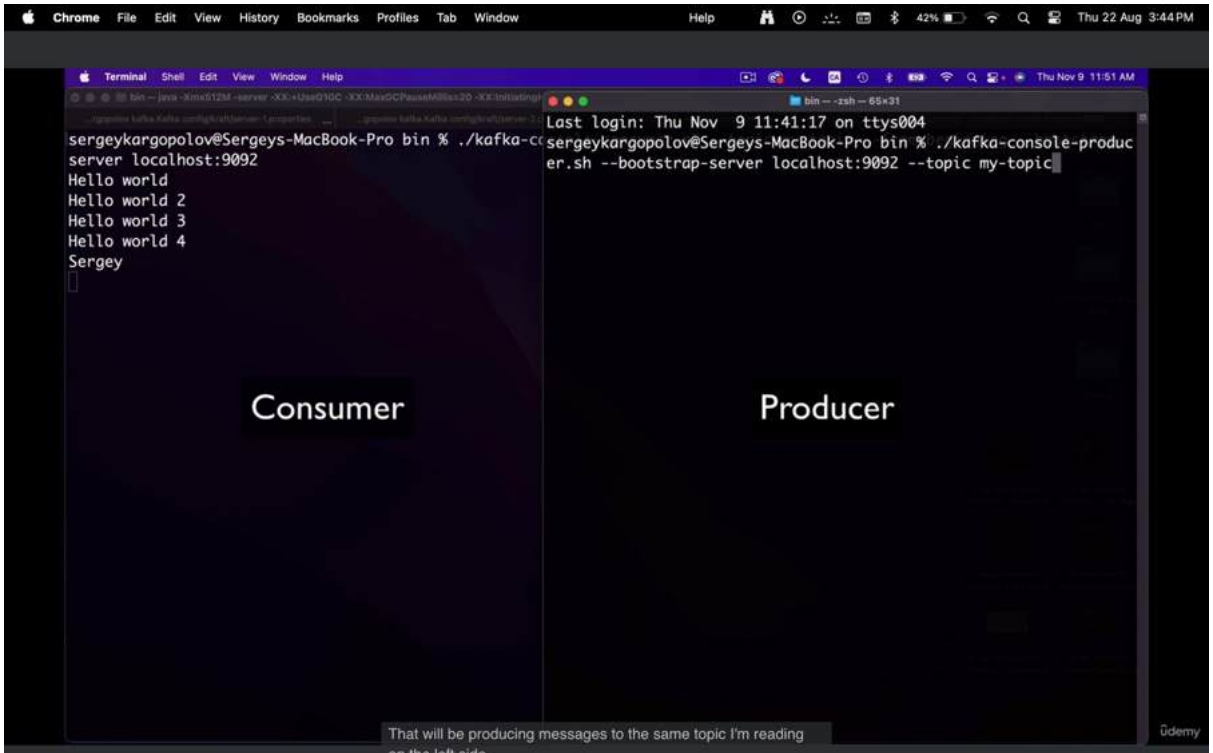
So let's continue.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-
server localhost:9092
```

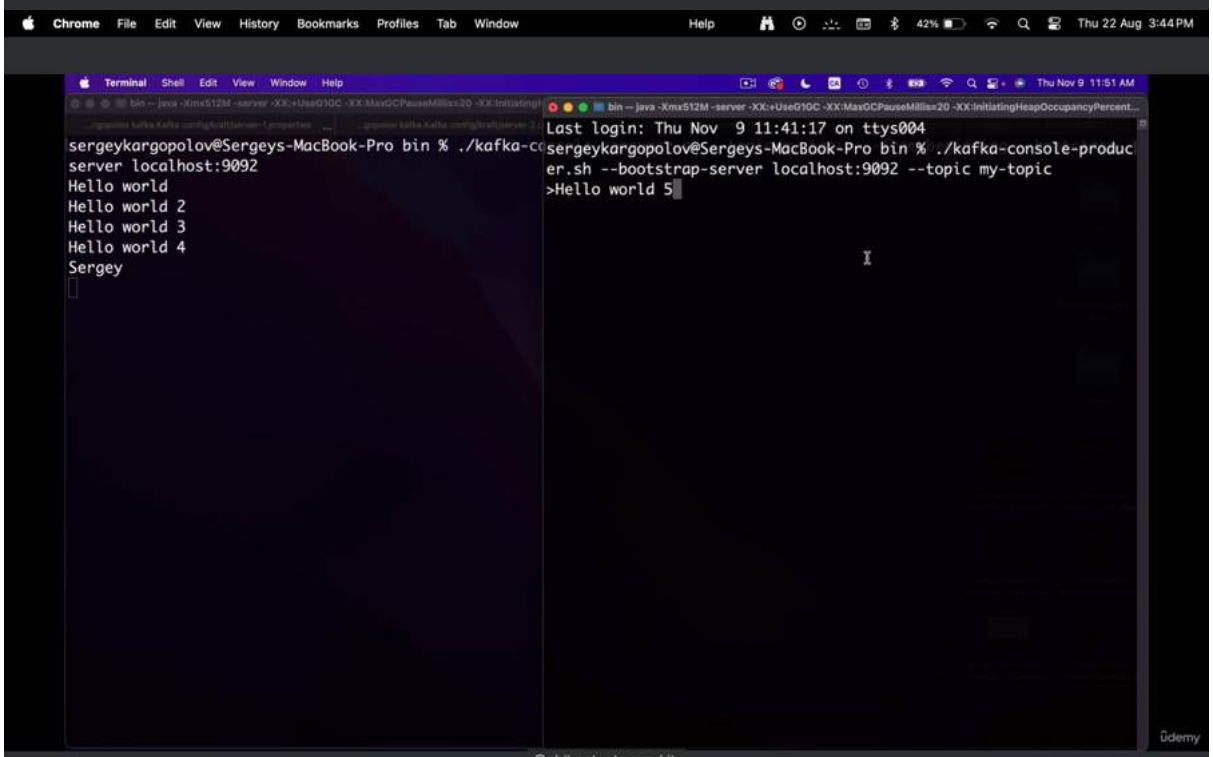
I will hit enter and we'll see if there are any messages in this topic.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
```

And here are the messages that we sent in previous video lessons.



That will be producing messages to the same topic I'm reading on the left side.



Or hit enter to send it.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-co
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>
```

And as soon as I hit enter and the message is sent to this Kafka topic, my consumer, running in a

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-co
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consum
er.sh --topic my-topic --from-beginning --bootstrap-server localh
ost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
```

consumer was able to read them as well.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Thu 22 Aug 3:45 PM

Terminal Shell Edit View Window Help Thu Nov 9 11:54 AM

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
```

Consumer

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
er.sh --topic my-topic --from-beginning --bootstrap-
ost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
```

Consumer

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>
```

Producer

And I will go back to the producer script and we'll try sending one more message to the same Kafka topic.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Thu 22 Aug 3:45 PM

Terminal Shell Edit View Window Help Thu Nov 9 11:55 AM

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
er.sh --topic my-topic --from-beginning --bootstrap-
ost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
>
```

Now notice that as soon as the message is sent to the topic, all consumers that are subscribed to read

31. Consuming messages from Kafka topic from the beginning

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
er.sh --topic my-topic --from-beginning --bootstrap-
ost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
```

Now notice that as soon as the message is sent to the topic, all consumers that are subscribed to read

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-s
erver localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-s
erver localhost:9092
```

If I run it again.

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
>

sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:9092
```

Now if I run the script again.

32. Consuming new Kafka messages only

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5

sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server loca
092
```

It is now connected to Kafka Cluster, but it did not read any old messages.

1.25x 1:19 / 2:00

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
```

```
Hello world 5
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-
console-consumer.sh --topic my-topic --from-beginni
ng --bootstrap-server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-
console-consumer.sh --topic my-topic --bootstrap-se
rver localhost:9092
Hello world 7
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produc
er.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
>Hello world 7
>
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic
 my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic
 my-topic --property "parse.key=true" --property "key.separator=:"
```

So this second parameter is to configure the separator between the key and the value.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-con
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
```

```
Hello world 4
Sergey
Hello world 5
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:9092
Hello world 7
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:9092
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --bootstrap-server localhost:9092 --topic my-topic --property "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-con
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
Sergey
```

```
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092
Hello world 7
sergeykargopolov@ onsumer.sh --topic my-topic --bootstrap-server localhost:
9092
Sergey
^CProcessed a tot
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092 print.key=t
```

Oops. It should be:
--property print.key=true

Equals true.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-cor
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
Sergey
Kargopolov
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic --prop
erty "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>lastName:Kargopolov
>
```

```
Hello world 6
^CProcessed a total of 7 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092
Hello world 7
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092
Sergey
^CProcessed a total of 1 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092 print.key=true
Kargopolov
^CProcessed a total of 1 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost:
9092 --property print.key=true
```

Okay, so if I repeat this command.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-cor
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
Sergey
Kargopolov
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic --prop
erty "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>lastName:Kargopolov
>lastName:Kargopolov
>
```

```
Hello world 7
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
onsole-consumer.sh --topic my-topic --bootstrap-serv
er localhost:9092
Sergey
^CProcessed a total of 1 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
onsole-consumer.sh --topic my-topic --bootstrap-serv
er localhost:9092 print.key=true
Kargopolov
^CProcessed a total of 1 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-c
onsole-consumer.sh --topic my-topic --bootstrap-serv
er localhost:9092 --property print.key=true
lastName        Kargopolov
```

This time I get a different response.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-cor
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
Sergey
Kargopolov
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic --prop
erty "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>lastName:Kargopolov
>lastName:Kargopolov
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost
:9092 --property print.key=true --property print.value=true --from-beginning
null    Hello world
null    Hello world 2
null    Hello world 3
null    Hello world 4
firstName       Sergey
null    Hello world 5
null    Hello world 6
null    Hello world 7
firstName       Sergey
lastName        Kargopolov
lastName        Kargopolov
```

And I see that those messages that were sent without a key are stored in Kafka topic with a null value

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-cor
server localhost:9092
Hello world
Hello world 2
Hello world 3
Hello world 4
Sergey
Hello world 5
Hello world 6
Hello world 7
Sergey
Kargopolov
```

```
Last login: Thu Nov  9 11:41:17 on ttys004
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic
>Hello world 5
>Hello world 6
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-produ
cer.sh --bootstrap-server localhost:9092 --topic my-topic --prop
erty "parse.key=true" --property "key.separator=:"
>firstName:Sergey
>lastName:Kargopolov
>lastName:Kargopolov
>
```

```
lastName        Kargopolov
^CProcessed a total of 11 messages
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic my-topic --bootstrap-server localhost
:9092 --property print.key=true --property print.value=false --from-beginning
null
null
null
null
firstName
null
null
null
firstName
lastName
lastName
```

And I see that message value is not printed.

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic messages-order --partitions 3 --replica
tion-factor 3 --bootstrap-server localhost:9092
Created topic messages-order.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --topic messages-order --bootstrap-server loca
lhost:9092 --property "parse.key=true" --property "key.separator=:"
>
```

These messages will need to have same ID, it can be the same user ID, or it can be the same product

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic messages-order --partitions 3 --replica
tion-factor 3 --bootstrap-server localhost:9092
Created topic messages-order.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --topic messages-order --bootstrap-server loca
lhost:9092 --property "parse.key=true" --property "key.separator=:"
>1:First message
>1:Second message
>
```

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic messages-order --partitions 3 --replica
tion-factor 3 --bootstrap-server localhost:9092
Created topic messages-order.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --topic messages-order --bootstrap-server loca
lhost:9092 --property "parse.key=true" --property "key.separator=:"
>1:First message
>1:Second message
>1:Third message
>1:Fourth message
>1:Fifth message
>1:Sixth message
>
```

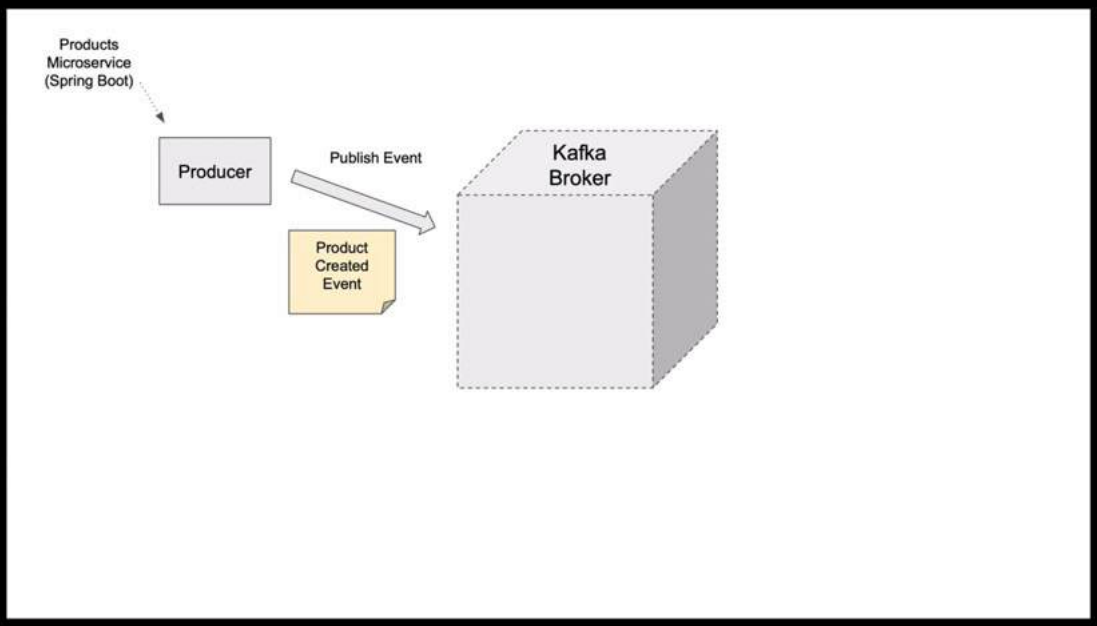
So now that I have a few messages with the same key, I will send a few messages, but all using a different

```
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-topics.sh --create --topic messages-order --partitions 3 --replica
tion-factor 3 --bootstrap-server localhost:9092
Created topic messages-order.
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-producer.sh --topic messages-order --bootstrap-server loca
lhost:9092 --property "parse.key=true" --property "key.separator=:"
>1:First message
>1:Second message
>1:Third message
>1:Fourth message
>1:Fifth message
>1:Sixth message
>a:a
>b:b
>c:c
>d:d
>e:e
>f:f
>g:g
>
```

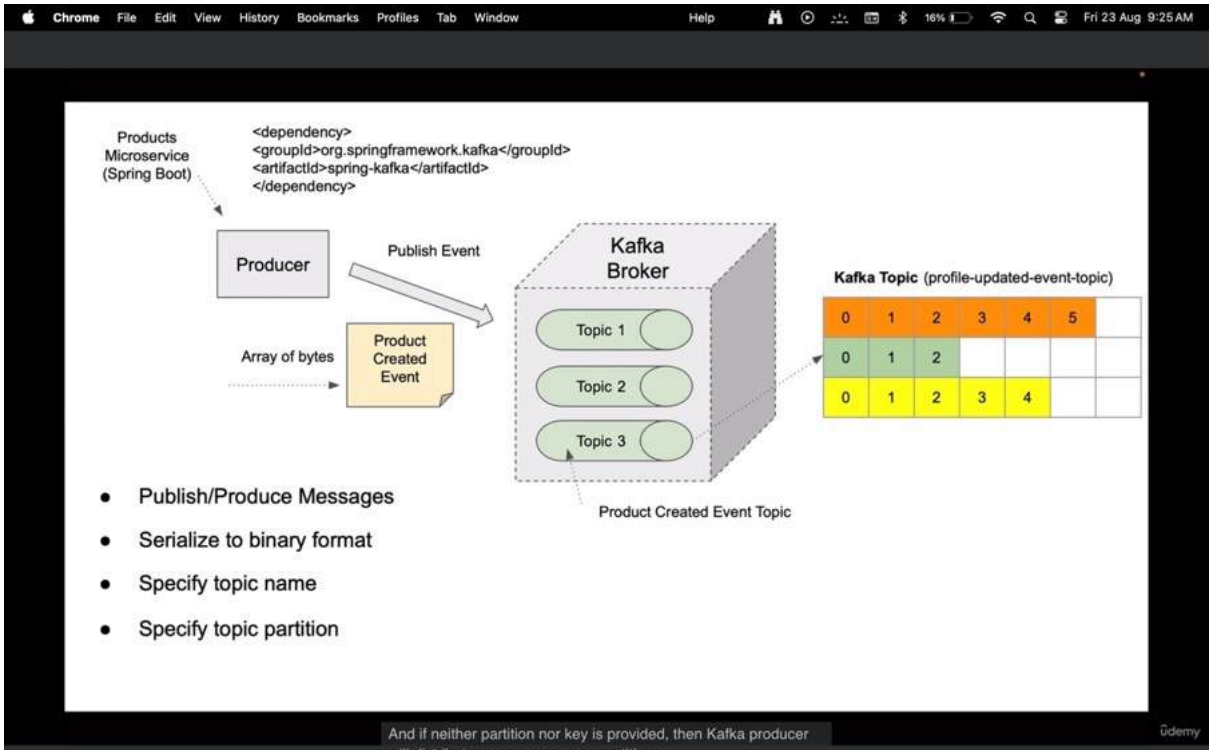
Notice that each message has a different key.

```
>1:First message
>1:Second message
>1:Third message
>1:Fourth message
>1:Fifth message
>1:Sixth message
>a:a
>b:b
>c:c
>d:d
>e:e
>f:f
>g:g
>^C%
sergeykargopolov@Sergeys-MacBook-Pro bin % ./kafka-console-consumer.sh --topic messages-order --bootstrap-server loca
lhost:9092 --from-beginning --property print.key=true
1       First message
1       Second message
1       Third message
1       Fourth message
1       Fifth message
1       Sixth message
f       f
g       g
a       a
c       c
b       b
d       d
e       e
```

Now notice that a list of messages with the same key.

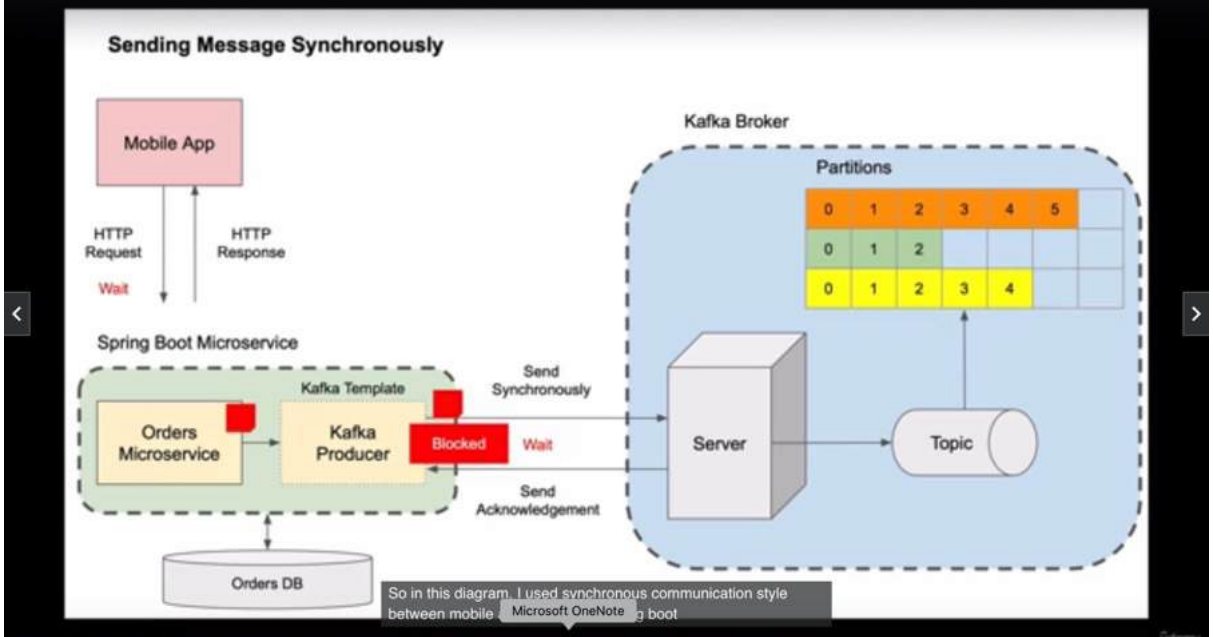


So the primary role of Kafka Producer is to publish messages or events to one or more Kafka topics.

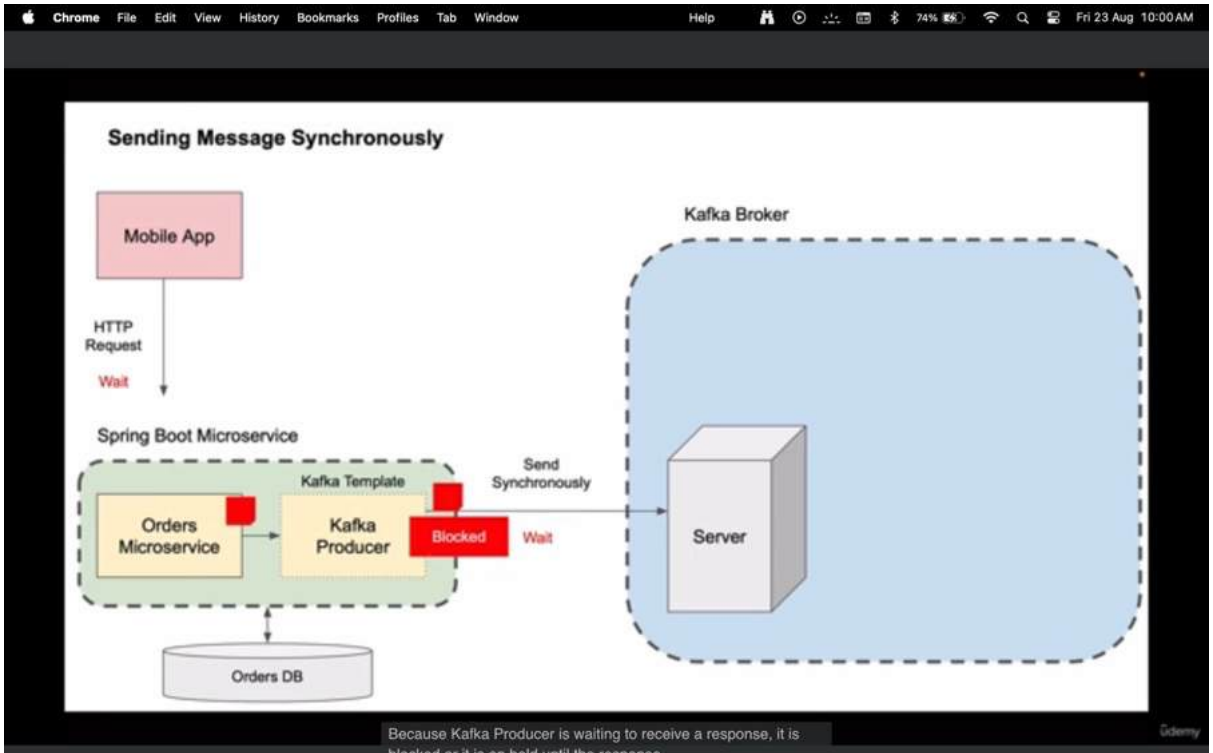


And if neither partition nor key is provided, then Kafka producer will distribute messages across partitions

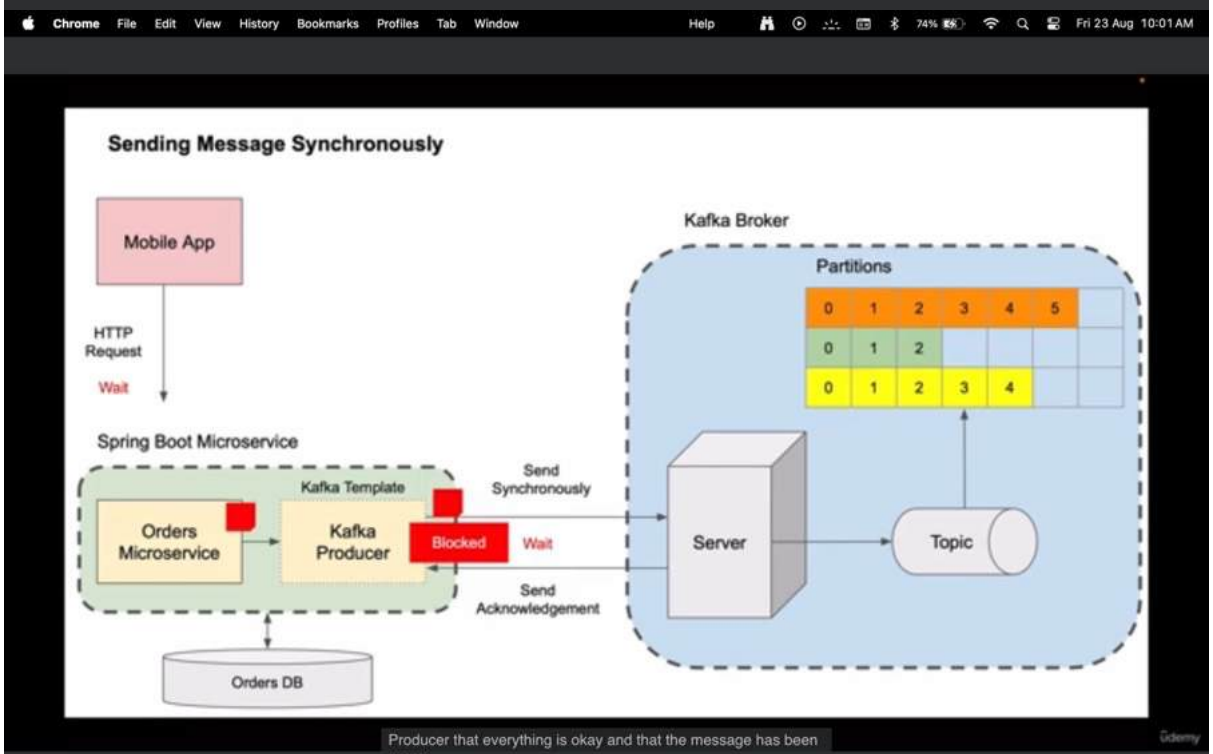
38. Kafka Producer - Introduction to synchronous communication style



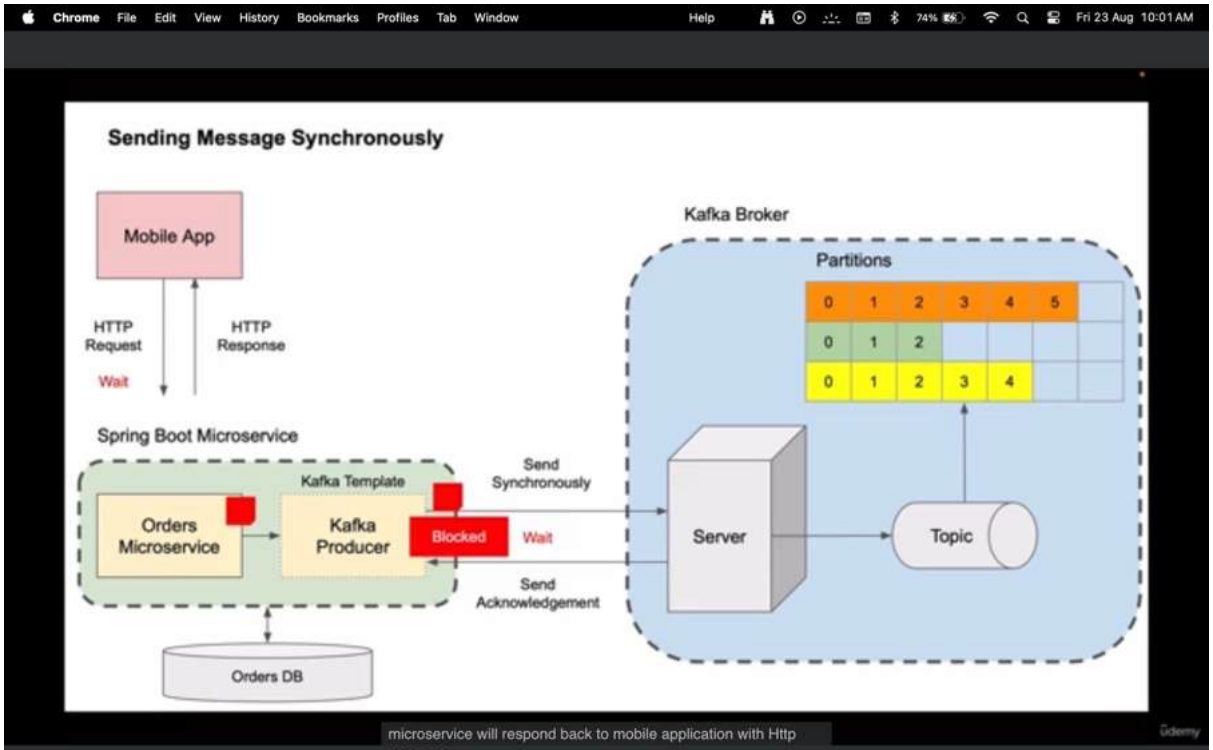
So in this diagram, I used synchronous communication style between mobile app and spring boot



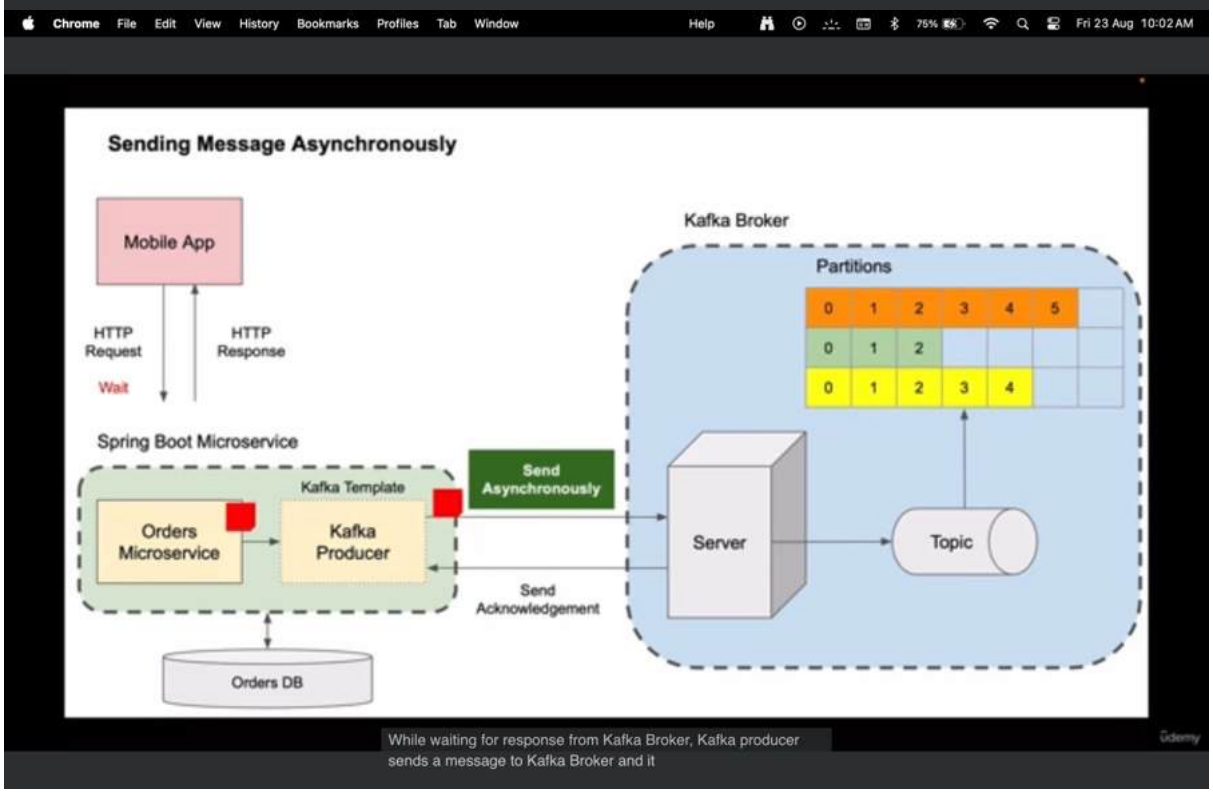
Because Kafka Producer is waiting to receive a response, it is blocked or it is on hold until the response



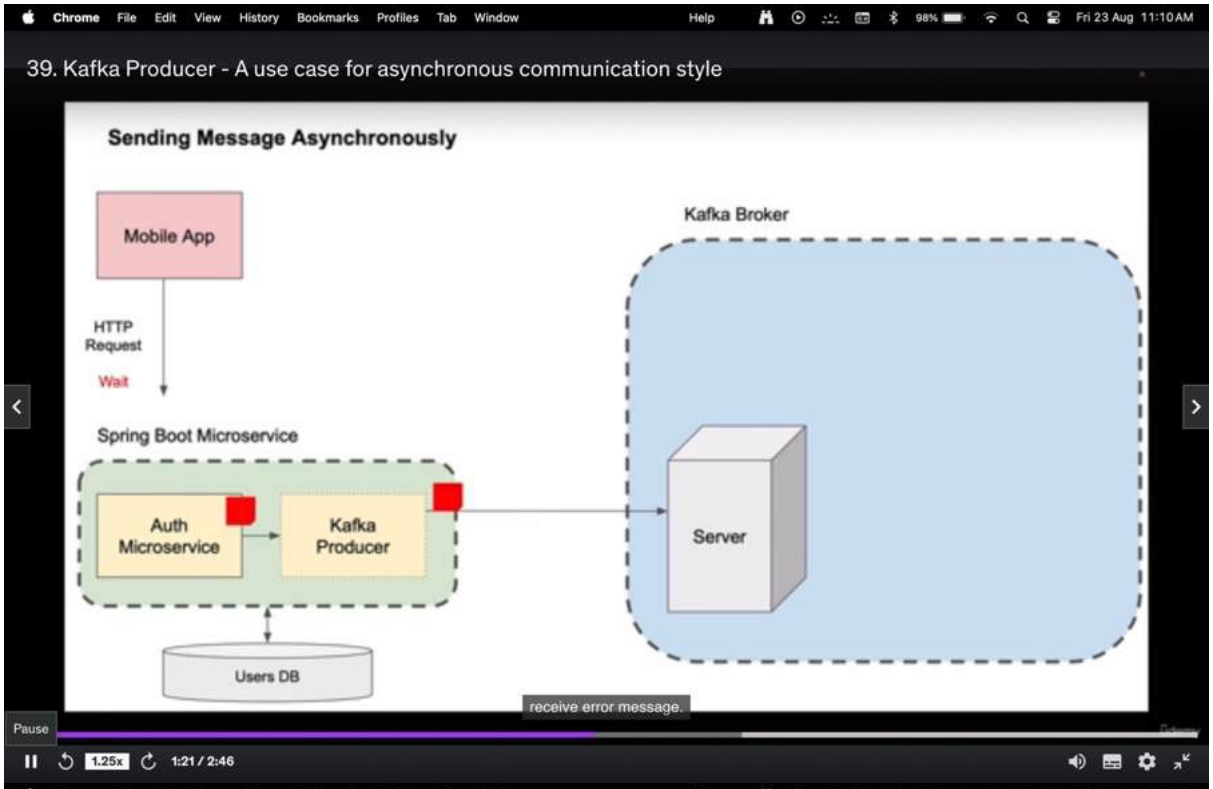
Producer that everything is okay and that the message has been stored.

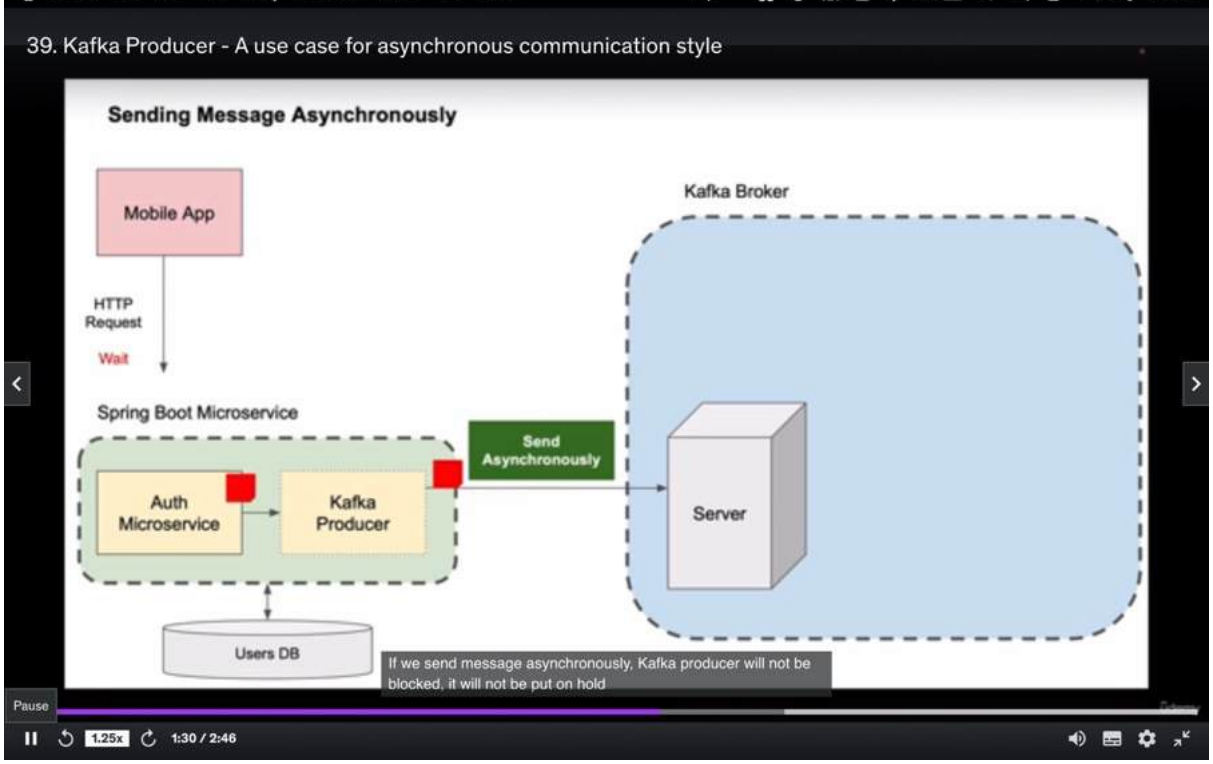


microservice will respond back to mobile application with Http response.

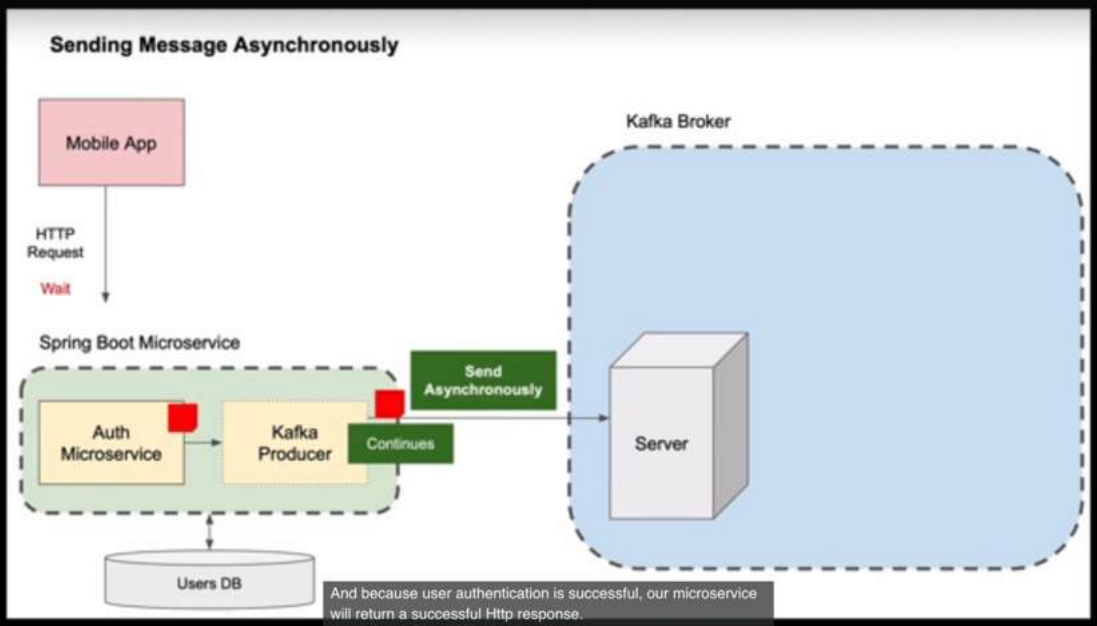


While waiting for response from Kafka Broker, Kafka producer sends a message to Kafka Broker and it

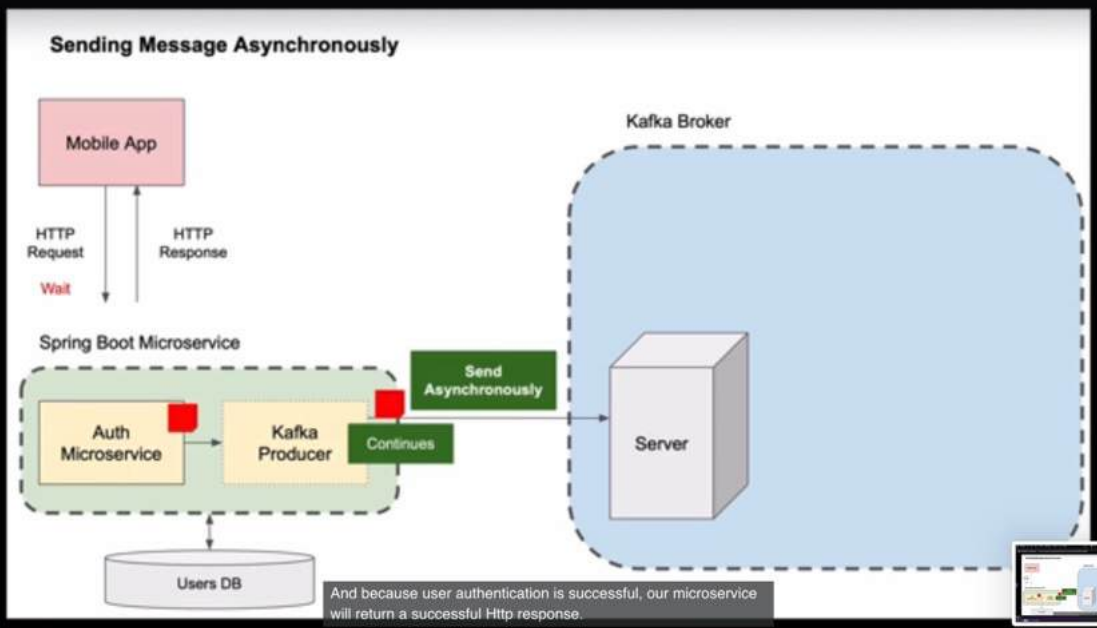

Chrome File Edit View History Bookmarks Profiles Tab Window Help 98% Fri 23 Aug 11:10 AM
39. Kafka Producer - A use case for asynchronous communication style
Sending Message Asynchronously
Mobile App
HTTP Request
Wait
Spring Boot Microservice
Auth Microservice
Kafka Producer
Users DB
Kafka Broker
Server
receive error message.
Pause
1.25x
1:21 / 2:46


Chrome File Edit View History Bookmarks Profiles Tab Window Help 98% Fri 23 Aug 11:10 AM
39. Kafka Producer - A use case for asynchronous communication style
Sending Message Asynchronously
Mobile App
HTTP Request
Wait
Spring Boot Microservice
Auth Microservice
Kafka Producer
Send Asynchronously
Users DB
Kafka Broker
Server
If we send message asynchronously, Kafka producer will not be blocked, it will not be put on hold
Pause
1.25x
1:30 / 2:46

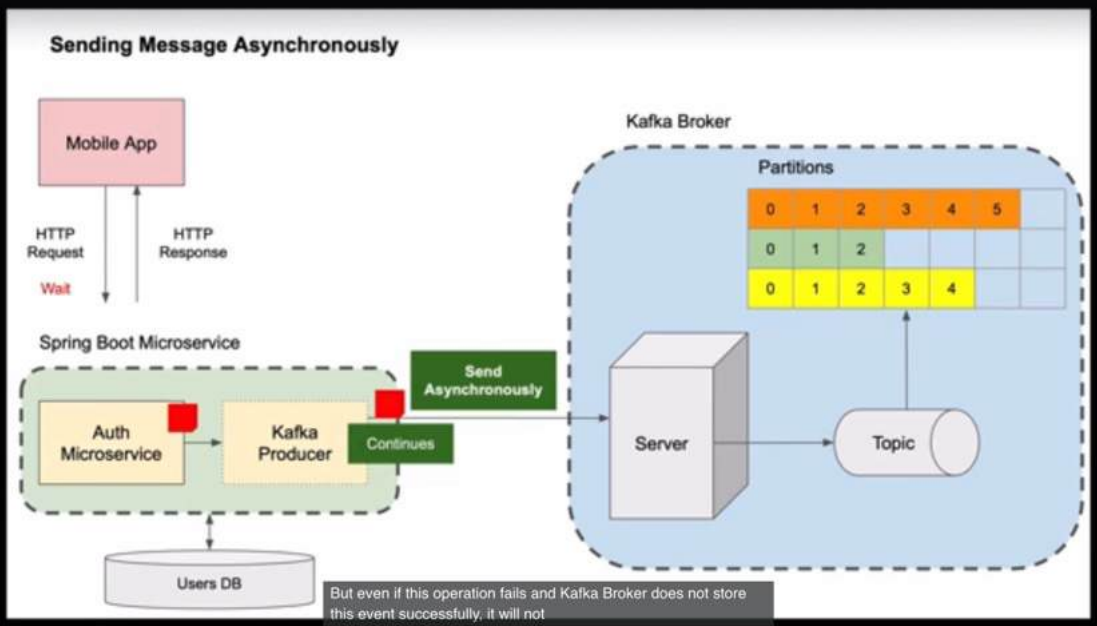
39. Kafka Producer - A use case for asynchronous communication style



39. Kafka Producer - A use case for asynchronous communication style

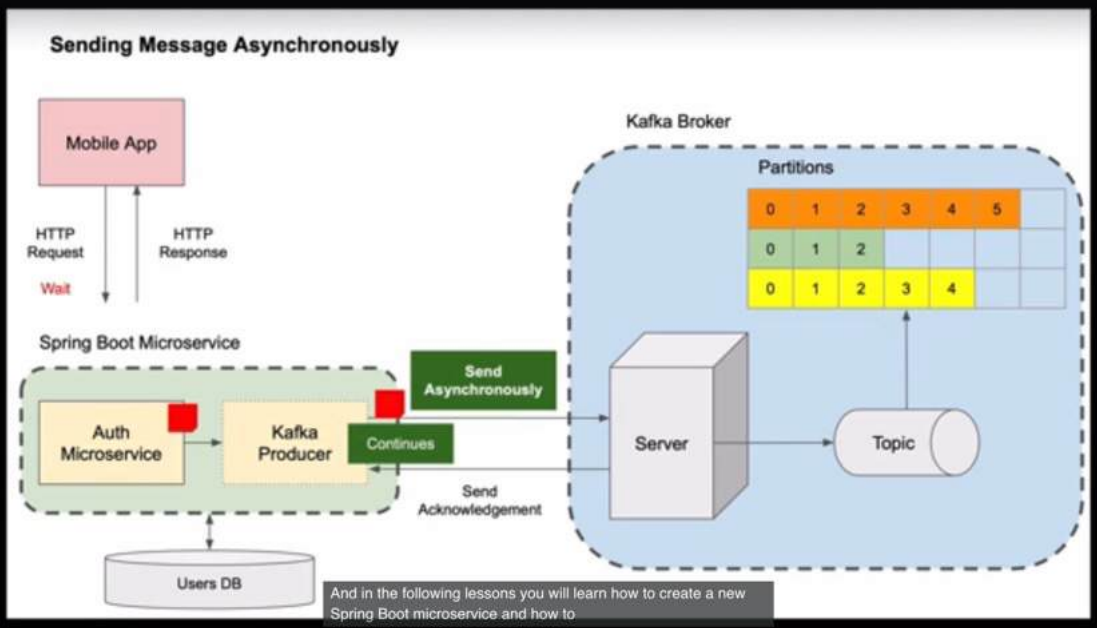


39. Kafka Producer - A use case for asynchronous communication style



Pause 1.25x 2:04 / 2:46

39. Kafka Producer - A use case for asynchronous communication style



Pause 1.25x 2:29 / 2:46

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSeria
```

org.apache.kafka.common.serialization.StringSerializer

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSerializer
```

org.springframework.kafka.support.serializer.JsonSerializer

Now notice that this time the serializer class comes from Spring Framework package rather than from

```java
package com.appsdeveloperblog.ws.products;

import org.apache.kafka.clients.admin.NewTopic;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.config.TopicBuilder;

@Configuration
public class KafkaConfig {

    NewTopic createTopic() {
        return TopicBuilder.name("product-created-events-topic")
                .partitions(3)
                .build();
    }

}
```

For example, if I create a topic with three partitions only like I do here, then I will be able to

```java
package com.appsdeveloperblog.ws.products;

import org.apache.kafka.clients.admin.NewTopic;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.config.TopicBuilder;

@Configuration
public class KafkaConfig {

    NewTopic createTopic() {
        return TopicBuilder.name("product-created-events-topic")
                .partitions(3)
                .replicas(3)
                .build();
    }

}
```

I'll go with three replicas as well.

42. Creating Kafka Topic

```java
package com.appsdeveloperblog.ws.products;

import java.util.Map;

import org.apache.kafka.clients.admin.NewTopic;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.config.TopicBuilder;

@Configuration
public class KafkaConfig {

    NewTopic createTopic() {
        return TopicBuilder.name("product-created-events-topic")
                .partitions(3)
                .replicas(3)
                .configs(Map.of("min.insync.replicas","2"))
                .build();
    }

}
```

It specifies the minimum number of replicas that must acknowledge the write operation for this write

1.25x 3:53 / 5:07

42. Creating Kafka Topic

```java
package com.appsdeveloperblog.ws.products;

import java.util.Map;

import org.apache.kafka.clients.admin.NewTopic;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.kafka.config.TopicBuilder;

@Configuration
public class KafkaConfig {

    @Bean
    NewTopic createTopic() {
        return TopicBuilder.name("product-created-events-topic")
                .partitions(3)
                .replicas(3)
                .configs(Map.of("min.insync.replicas","2"))
                .build();
    }

}
```

So this bin annotation, it will add the new topic object that is created and returned from this method

1.25x 4:46 / 5:07

43. Run Microservice to create a new Kafka topic

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-topics.sh --bootstra-server localhost:9092 --describe --topi
c product-created-events-topic
Exception in thread "main" joptsimple.UnrecognizedOptionException: bootstra-server is not a recognized option
        at joptsimple.OptionException.unrecognizedOption(OptionException.java:108)
        at joptsimple.OptionParser.handleLongOptionToken(OptionParser.java:510)
        at joptsimple.OptionParserState$2.handleArgument(OptionParserState.java:56)
        at joptsimple.OptionParser.parse(OptionParser.java:396)
        at kafka.admin.TopicCommand$TopicCommandOptions.<init>(TopicCommand.scala:558)
        at kafka.admin.TopicCommand$.main(TopicCommand.scala:49)
        at kafka.admin.TopicCommand.main(TopicCommand.scala)
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-topics.sh --bootstrap-server localhost:9092 --describe --top
ic product-created-events-topic
Topic: product-created-events-topic     TopicId: 7IR-i17USR6GDAMmZTd89Q PartitionCount: 3       ReplicationFactor: 3C
onfigs: min.insync.replicas=2,segment.bytes=1073741824
        Topic: product-created-events-topic     Partition: 0    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: product-created-events-topic     Partition: 1    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: product-created-events-topic     Partition: 2    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

And here I have a very detailed response.

1.25x 1:53 / 3:08

```
package com.appsdeveloperblog.ws.products.rest;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    @PostMapping
    public ResponseEntity<String> createProduct() {
        return ResponseEntity.status(HttpStatus.CREATED).body("");
    }

}
```

And in the request body, it will send a Json document containing product details.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/ProductController.java - Spring Tool Suite 4

KafkaConfig.java | *ProductController.java

```java
package com.appsdeveloperblog.ws.products.rest;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    @PostMapping
    public ResponseEntity<String> createProduct(@RequestBody CreateProductRestModel product) {
        return ResponseEntity
    }

}
```

CreateProductRestModel cannot be resolved to a type

7 quick fixes available:

- Create class 'CreateProductRestModel'
- Create record 'CreateProductRestModel'
- Create interface 'CreateProductRestModel'
- Create enum 'CreateProductRestModel'
- Add type parameter 'CreateProductRestModel' to 'ProductController'
- Add type parameter 'CreateProductRestModel' to 'createProduct(CreateProductRestModel)'
- Fix project setup...

I will move my mouse over the create product model class and will choose option to create class.

44. Creating Rest Controller

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/CreateProductRestModel.java - Spring Tool Suite 4

KafkaConfig.java | *ProductController.java | CreateProductRestModel.java

```java
package com.appsdeveloperblog.ws.products.rest;

import java.math.BigDecimal;

public class CreateProductRestModel {

    private String title;
    private BigDecimal price;
    private Integer quantity;

    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public BigDecimal getPrice() {
        return price;
    }
    public void setPrice(BigDecimal price) {
        this.price = price;
    }
    public Integer getQuantity() {
        return quantity;
    }
    public void setQuantity(Integer quantity) {
        this.quantity = quantity;
    }


}
```

Let's close it and let's go back to our products controller.

1.25x 4:26 / 5:24

```java
package com.appsdeveloperblog.ws.products.rest;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    @PostMapping
    public ResponseEntity<String> createProduct(@RequestBody CreateProductRestModel product) {
        return ResponseEntity.status(HttpStatus.CREATED).body("");
    }

}
```

```java
package com.appsdeveloperblog.ws.products.rest;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.appsdeveloperblog.ws.products.service.ProductService;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    ProductService productService;

    public ProductController(ProductService productService) {
        this.productService = productService;
    }

    @PostMapping
    public ResponseEntity<String> createProduct(@RequestBody CreateProductRestModel product

        String productId = productService.createProduct(product);

        return ResponseEntity.status(HttpStatus.CREATED).body(productId);
    }

}
```

All right so let's save changes in our rest controller class.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

```java
package com.appsdeveloperblog.ws.products.service;

import org.springframework.stereotype.Service;

@Service
public class ProductServiceImpl implements ProductService {

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {
        // TODO Auto-generated method stub
        return null;
    }

}
```

So in this lesson we will create a new class that will be used to create event object.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductCreatedEvent.java - Spring Tool Suite 4

```java
public class ProductCreatedEvent {

    private String productId;
    private String title;
    private BigDecimal price;
    private Integer quantity;

    public ProductCreatedEvent() {

    }

    public ProductCreatedEvent(String productId, String title, BigDecimal price, Integer qu
        this.productId = productId;
        this.title = title;
        this.price = price;
        this.quantity = quantity;
    }

    public String getProductId() {
        return productId;
    }

    public void setProductId(String productId) {
        this.productId = productId;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public BigDecimal getPrice() {
        return price;
```

And this is because when this event object is deserialized by the consumer microservice, the Deserialization

```
String productId = UUID.randomUUID().toString();

// TODO: Persist Product Details into database tab

ProductCreatedEvent productCreatedEvent = new Prod
        productRestModel.getTitle(), productRestMo
        productRestModel.getQuantity());
```

nformation in message headers.
converted before sending.
aOperations

send.

ult.

```
send
send(Message<?> message) : CompletableFuture<Send
send(ProducerRecord<String,ProductCreatedEvent> rec
send(String topic, ProductCreatedEvent data) : Complet
send(String topic, String key, ProductCreatedEvent data
send(String topic, Integer partition, String key, ProductC
send(String topic, Integer partition, Long timestamp, Str
sendDefault(ProductCreatedEvent data) : CompletableF
sendDefault(String key, ProductCreatedEvent data) : Co
sendDefault(Integer partition, String key, ProductCreate
sendDefault(Integer partition, Long timestamp, String ke
```

from proposal table or click Kafka template provides a number of send methods that can be used to send messages to Kafka topic, ce' to show Template Proposals

30 : 27


```
String productId = UUID.randomUUID().toString();

// TODO: Persist Product Details into database tab

ProductCreatedEvent productCreatedEvent = new Prod
        productRestModel.getTitle(), productRestMo
        productRestModel.getQuantity());
```

nformation in message headers.
converted before sending.
aOperations

send.

ult.

```
send
send(Message<?> message) : CompletableFuture<Send
send(ProducerRecord<String,ProductCreatedEvent> rec
send(String topic, ProductCreatedEvent data) : Complet
send(String topic, String key, ProductCreatedEvent data
send(String topic, Integer partition, String key, ProductC
send(String topic, Integer partition, Long timestamp, Str
sendDefault(ProductCreatedEvent data) : CompletableF
sendDefault(String key, ProductCreatedEvent data) : Co
sendDefault(Integer partition, String key, ProductCreate
sendDefault(Integer partition, Long timestamp, String ke
```

from proposal table or click I will use one that accepts topic name, message key, and event object. ce' to show Template Proposals

30 : 27

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);


        return null;
    }

}
```

Now this will send a message to Kafka topic as synchronously, and it will not wait for acknowledgement

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);


        return productId;
    }

}
```

And my controller class will also return this product ID in response body.

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);


        return productId;
    }

}
```

So if you want to send message asynchronously and not wait for a result, then this is all you need

```java
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.kafka.support.SendResult;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);



        return productId;
    }

}
```

So this Completablefuture class is a class in Java concurrency API, and it represents a future result

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEvent.java | *ProductServiceImpl.java | KafkaConfig.java

```java
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);

        future.whenComplete((result, exception) -> {

            if(exception != null) {
                LOGGER.error("Failed to send message: " + exception.getMessage());
            } else {
                LOGGER.info("Message sent successfully: " + result.getRecordMetadata());
            }

        });


        return productId;
    }

}
```

Then it will respond back with a successful response as well.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEvent.java | *ProductServiceImpl.java | KafkaConfig.java

```java
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);

        future.whenComplete((result, exception) -> {

            if(exception != null) {
                LOGGER.error("Failed to send message: " + exception.getMessage());
            } else {
                LOGGER.info("Message sent successfully: " + result.getRecordMetadata());
            }

        });

        future.join();

        return productId;
    }

}
```

So basically it will become synchronous operation.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEvent.java | *ProductServiceImpl.java | KafkaConfig.java

```java
    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);

        future.whenComplete((result, exception) -> {

            if(exception != null) {
                LOGGER.error("****** Failed to send message: " + exception.getMessage());
            } else {
                LOGGER.info("******* Message sent successfully: " + result.getRecordMetadata());
            }

        });

        LOGGER.info("***** Returning product id");

        return productId;
    }

}
```

the result, then execution will continue and the returning product ID log message should be displayed

48. Kafka Asynchronous Send. Trying how it works.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic
```

Pause

1.25x 1:52 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic product-created-events-topic --b
ootstrap-server localhost:9092 --property print.key=true
```

Now let's run this command.

Pause

1.25x 2:12 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

```
17      KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
18      private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());
19
20      public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
21          this.kafkaTemplate = kafkaTemplate;
22      }
23
24      @Override
25      public String createProduct(CreateProductRestModel productRestModel) {
26
27          String productId = UUID.randomUUID().toString();
28
29          // TODO: Persist Product Details into database table before publishing an Event
30
31          ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
32                  productRestModel.getTitle(), productRestModel.getPrice(),
33                  productRestModel.getQuantity());
```

```
ssl.trustmanager.algorithm = PKIX
ssl.truststore.certificates = null
ssl.truststore.location = null
ssl.truststore.password = null
ssl.truststore.type = JKS

2023-11-28T11:24:36.110-05:00  INFO 59157 --- [           main] o.a.kafka
2023-11-28T11:24:36.111-05:00  INFO 59157 --- [           main] o.a.kafka.co
2023-11-28T11:24:36.111-05:00  INFO 59157 --- [           main] o.a.kafka.comm
2023-11-28T11:24:36.366-05:00  INFO 59157 --- [] admin client-1] o.a.kafka.common
2023-11-28T11:24:36.371-05:00  INFO 59157 --- [] admin client-1] o.apache.kafka.com
2023-11-28T11:24:36.371-05:00  INFO 59157 --- [] admin client-1] o.apache.kafka.comm
2023-11-28T11:24:36.371-05:00  INFO 59157 --- [] admin client-1] o.apache.kafka.common
```

My products microservice application is running, and now I will open postman to send the Http request

Pause

1.25x 2:45 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

```
KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
    this.kafkaTemplate = kafkaTemplate;
}

@Override
public String createProduct(CreateProductRestModel productRestModel) {

    String productId = UUID.randomUUID().toString();

    // TODO: Persist Product Details into database table before publishing an Event

    ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
            productRestModel.getTitle(), productRestModel.getPrice(),
            productRestModel.getQuantity());
```

```
mon.utils.AppInfoParser     : Kafka version: 3.6.0
mon.utils.AppInfoParser     : Kafka commitId: 60e845626d8a465a
mon.utils.AppInfoParser     : Kafka startTimeMs: 1701188676109
mon.utils.AppInfoParser     : App info kafka.admin.client for adminclient-1 unregistered
).common.metrics.Metrics    : Metrics scheduler closed
).common.metrics.Metrics    : Closing reporter org.apache.kafka.common.metrics.JmxReporter
).common.metrics.Metrics    : Metrics reporters closed
ded.tomcat.TomcatWebServer  : Tomcat started on port [redacted] (http) with context path ''
:tsMicroserviceApplication  : Started ProductsMicroserviceApplication in 1.725 seconds (process r
```

So I will copy port number and we'll paste it here.

Pause 1.25x 3:21 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

POST http://localhost:58806/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Response

So let's try sending this request and see if it works.

Pause 1.25x 4:04 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

POST http://localhost:58806/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 236 ms Size: 205 B

```
99f49d55-28d1-4fe3-ad61-1b166fcd56cf
```

And we have a successful response.

4:10 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic product-created-events-topic --b
ootstrap-server localhost:9092 --property print.key=true
99f49d55-28d1-4fe3-ad61-1b166fcd56cf    {"productId":"99f49d55-28d1-4fe3-ad61-1b166fcd56cf","title":"iPhone 11","pric
e":800,"quantity":19}
```

And I see that my Kafka consumer did receive one message.

4:22 / 5:40

48. Kafka Asynchronous Send. Trying how it works.

```
KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
    this.kafkaTemplate = kafkaTemplate;
}

@Override
public String createProduct(CreateProductRestModel productRestModel) {

    String productId = UUID.randomUUID().toString();

    // TODO: Persist Product Details into database table before publishing an Event

    ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
            productRestModel.getTitle(), productRestModel.getPrice(),
            productRestModel.getQuantity());
```

```
.support.serializer.JsonSerializer

ec-2] o.a.k.clients.producer.KafkaProducer     : [Producer clientId=producer-1] Instantiated an idempotent producer.
ec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka version: 3.6.0
ec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka commitId: 60e845626d8a465a
ec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka startTimeMs: 1701188950594
er-1] org.apache.kafka.clients.Metadata        : [Producer clientId=producer-1] Cluster ID: 0BW7jPL6QUqypgM7AVt7gg
ec-2] c.a.w.p.service.ProductServiceImpl       : ***** Returning product id
er-1] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-1] ProducerId set to 1000 with epoch 0
er-1] c.a.w.p.service.ProductServiceImpl       : ****** Message sent successfully: product-created-events-topic-1@0
```

1.25x 5:07 / 5:40

49. Kafka Producer: Send Message Synchronously

```
KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
    this.kafkaTemplate = kafkaTemplate;
}

@Override
public String createProduct(CreateProductRestModel productRestModel) {

    String productId = UUID.randomUUID().toString();

    // TODO: Persist Product Details into database table before publishing an Event

    ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
            productRestModel.getTitle(), productRestModel.getPrice(),
            productRestModel.getQuantity());

    CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
            kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);

    future.whenComplete((result, exception) -> {

        if(exception != null) {
            LOGGER.error("****** Failed to send message: " + exception.getMessage());
        } else {
            LOGGER.info("******* Message sent successfully: " + result.getRecordMetadata());
        }

    });

    LOGGER.info("***** Returning product id");

    return productId;
}
}
```

If I send Kafka message synchronously, then I can configure my microservice application to wait for

1.25x 0:06 / 4:50

49. Kafka Producer: Send Message Synchronously

*ProductServiceImpl.java

```java
    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        CompletableFuture<SendResult<String, ProductCreatedEvent>> future =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent);

        future.whenComplete((result, exception) -> {

            if(exception != null) {
                LOGGER.error("****** Failed to send message: " + exception.getMessage());
            } else {
                LOGGER.info("******* Message sent successfully: " + result.getRecordMetadata());
            }

        });

        future.join();

        LOGGER.info("***** Returning product id");

        return productId;
    }
```

With this line of code, even though the Completablefuture is used.

1.25x 0:46 / 4:50

49. Kafka Producer: Send Message Synchronously

*ProductServiceImpl.java

```java
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.kafka.support.SendResult;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("***** Returning product id");

        return productId;
    }

}
```

So before using this semicolon I will call get method on it and then close this line.

1.25x 2:51 / 4:50

49. Kafka Producer: Send Message Synchronously

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

*ProductServiceImpl.java

```java
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.kafka.support.SendResult;
import org.springframework.stereotype.Service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("***** Returning product id");

        return productId;
    }

}
```

And because my method signature changed, I will also need to update it in the interface.

1.25x 3:37 / 4:50

49. Kafka Producer: Send Message Synchronously

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductService.java - Spring Tool Suite 4

Package Explorer

- ProductsMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.produ
      - KafkaConfig.java
      - ProductsMicroserviceApplicatio
    - com.appsdeveloperblog.ws.produ
      - CreateProductRestModel.java
      - ProductController.java
    - com.appsdeveloperblog.ws.produ
      - ProductCreatedEvent.java
      - ProductService.java
      - ProductServiceImpl.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

*ProductServiceImpl.java | ProductService.java

```java
package com.appsdeveloperblog.ws.products.service;

import com.appsdeveloperblog.ws.products.rest.CreateProductRestModel;

public interface ProductService {

    String createProduct(CreateProductRestModel productRestModel) throws Exception ;

}
```

Now let's save changes and go back to product service implementation.

1.25x 3:52 / 4:50

Chrome File Edit View History Bookmarks Profiles Tab Window Help 22% Sat 24 Aug 3:45 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Tue Nov 28 6:37 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/ErrorMessage.java - Spring Tool Suite 4

Package Explorer

- ProductsMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.produ
      - KafkaConfig.java
      - ProductsMicroserviceApplicatio
    - com.appsdeveloperblog.ws.produ
      - CreateProductRestModel.java
      - ErrorMessage.java
      - ProductController.java
    - com.appsdeveloperblog.ws.produ
      - ProductCreatedEvent.java
      - ProductService.java
      - ProductServiceImpl.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

ProductServiceImpl.java | ProductService.java | *ProductController.java | *ErrorMessage.java

```
    }

    public ErrorMessage(Date timestamp, String message, String details) {
        super();
        this.timestamp = timestamp;
        this.message = message;
        this.details = details;
    }

    public Date getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(Date timestamp) {
        this.timestamp = timestamp;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public String getDetails() {
        return details;
    }

    public void setDetails(String details) {
        this.details = details;
    }


}
```

Writable Smart Insert 47 : 6 : 763

Udemy

Let's save changes in error message class.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 22% Sat 24 Aug 3:46 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Tue Nov 28 6:40 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/ProductController.java - Spring Tool Suite 4

Package Explorer

- ProductsMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.produ
      - KafkaConfig.java
      - ProductsMicroserviceApplicatio
    - com.appsdeveloperblog.ws.produ
      - CreateProductRestModel.java
      - ErrorMessage.java
      - ProductController.java
    - com.appsdeveloperblog.ws.produ
      - ProductCreatedEvent.java
      - ProductService.java
      - ProductServiceImpl.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

ProductServiceImpl.java | ProductService.java | *ProductController.java | ErrorMessage.java

```
package com.appsdeveloperblog.ws.products.rest;

import java.util.Date;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    ProductService productService;

    public ProductController(ProductService productService) {
        this.productService = productService;
    }

    @PostMapping
    public ResponseEntity<Object> createProduct(@RequestBody CreateProductRestModel product) {

        String productId;
        try {
            productId = productService.createProduct(product);
        } catch (Exception e) {
            e.printStackTrace();
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR)
                    .body(new ErrorMessage(new Date(), e.getMessage(),"/products"));
        }

        return ResponseEntity.status(HttpStatus.CREATED).body(productId);
    }

}
```

1 : 85 : 1107

Udemy

This way, if exception takes place in our create product method, we will return error message.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/ProductController.java - Spring Tool Suite 4

```java
import java.util.Date;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.appsdeveloperblog.ws.products.service.ProductService;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    ProductService productService;
    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    public ProductController(ProductService productService) {
        this.productService = productService;
    }

    @PostMapping
    public ResponseEntity<Object> createProduct(@RequestBody CreateProductRestModel product) {

        String productId;
        try {
            productId = productService.createProduct(product);
        } catch (Exception e) {
            e.printStackTrace();
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR)
                    .body(new ErrorMessage(new Date(), e.getMessage(),"/products"));
        }
```

And now we can use this logger instead of print stack trace.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/rest/ProductController.java - Spring Tool Suite 4

```java
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.appsdeveloperblog.ws.products.service.ProductService;

@RestController
@RequestMapping("/products") //http://localhost:<port>/products
public class ProductController {

    ProductService productService;
    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    public ProductController(ProductService productService) {
        this.productService = productService;
    }

    @PostMapping
    public ResponseEntity<Object> createProduct(@RequestBody CreateProductRestModel product) {

        String productId;
        try {
            productId = productService.createProduct(product);
        } catch (Exception e) {
            //e.printStackTrace();
            LOGGER.error(e.getMessage(), e);
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR)
                    .body(new ErrorMessage(new Date(), e.getMessage(),"/products"));
        }

        return ResponseEntity.status(HttpStatus.CREATED).body(productId);
    }

}
```

And now that we're using logger to log error message if needed, you can turn it off using application

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

*ProductServiceImpl.java × | ProductService.java | ProductController.java | ErrorMessage.java

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("**** Returning product id");

        return productId;
    }

}
```

The value of the local variable result is not used

Then once the send method is called, execution will wait for it to receive the result, and once the

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

*ProductServiceImpl.java × | ProductService.java | ProductController.java | ErrorMessage.java

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRest                    s Exception {

        String productId = UUID.randomUUID().toSt

        // TODO: Persist Product Details into dat                    an Event

        ProductCreatedEvent productCreatedEvent =                    uctId,
                productRestModel.getTitle(), prod
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("**** Returning product id");

        return productId;
    }

}
```

- Topic name,
- Partition,
- Offset,
- Timestamp...

The value of the local variable result is not used

Topic name we can read from it.

51. Kafka Producer: Logging Record Metadata Information

*ProductServiceImpl.java × ProductService.java ProductController.java ErrorMessage.java

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());

        LOGGER.info("***** Returning product id");

        return productId;
    }
```

Okay, so this will print the topic name the event was published to and I can also print offset.

1.25x 1:54 / 2:22

51. Kafka Producer: Logging Record Metadata Information

*ProductServiceImpl.java × ProductService.java ProductController.java ErrorMessage.java

```java
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEvent).get();

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());
        LOGGER.info("Offset: " + result.getRecordMetadata().offset());

        LOGGER.info("***** Returning product id");

        return productId;
```

All right, so with this new line, I'm printing the offset of the record in the topic partition, some

1.25x 2:09 / 2:22

52. Kafka Synchronous Send. Trying how it works.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic product-created-events-topic --b
ootstrap-server localhost:9092 --property print.key=true
```

And now that my Kafka console consumer script is running, I will open postman and I will create one

1.25x 1:09 / 2:45

52. Kafka Synchronous Send. Trying how it works.

POST http://localhost:63081/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 236 ms Size: 205 B

```
99f49d55-28d1-4fe3-ad61-1b166fcd56cf
```

Let's go back to postman and use the correct port number.

1.25x 1:36 / 2:45

52. Kafka Synchronous Send. Trying how it works.

POST http://localhost:63081/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 596 ms Size: 205 B

```
ab069d90-7170-468c-813b-1f6dbc41b062
```

And we have a successful response.

1:51 / 2:45

52. Kafka Synchronous Send. Trying how it works.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic product-created-events-topic --b
ootstrap-server localhost:9092 --property print.key=true
ab069d90-7170-468c-813b-1f6dbc41b062    {"productId":"ab069d90-7170-468c-813b-1f6dbc41b062","title":"iPhone 11","pric
e":800,"quantity":19}
```

And here in the Kafka consumer script, I see that it was able to consume published event.

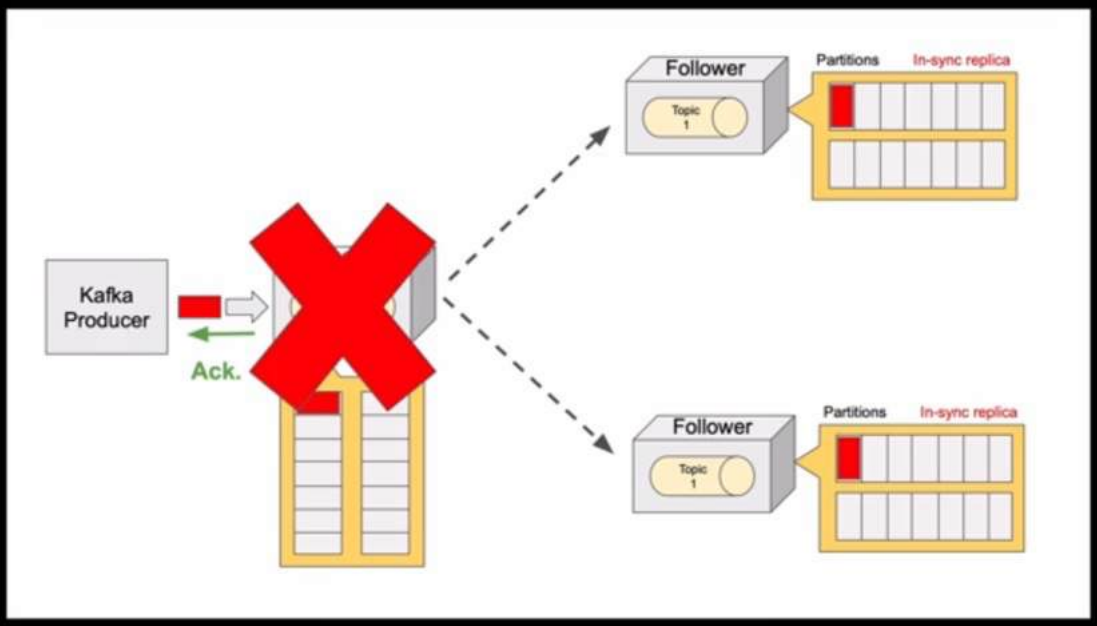
2:07 / 2:45

52. Kafka Synchronous Send. Trying how it works.

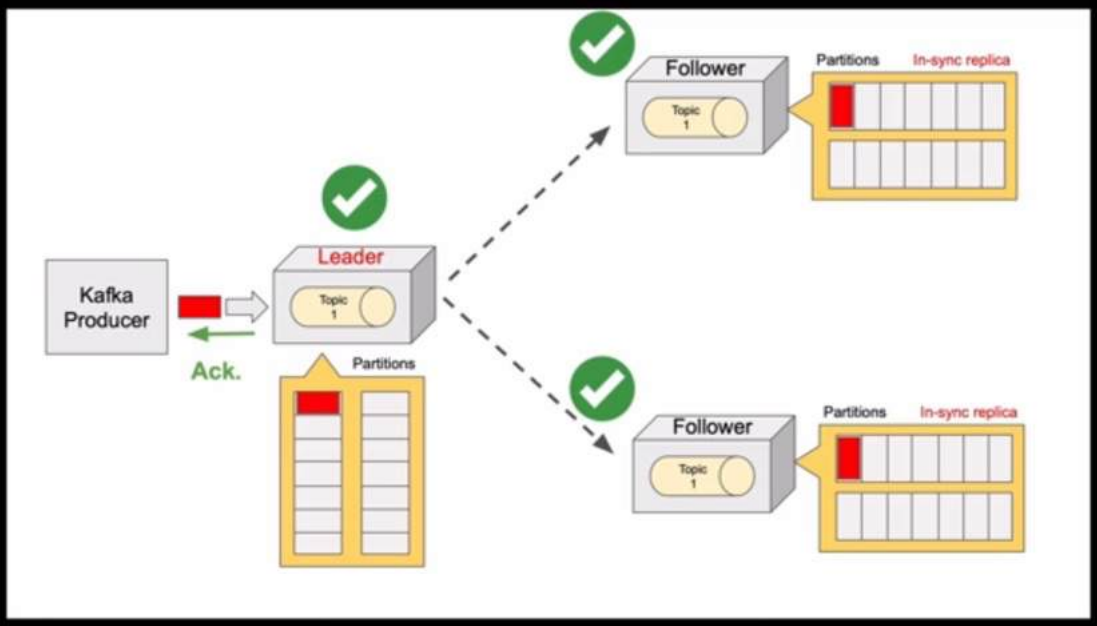
```
ole ×  Progress
t App] /Library/Java/JavaVirtualMachines/jdk-17.0.4.jdk/Contents/Home/bin/java  (Nov. 29, 2023, 3:55:35 p.m.) [pid: 970]

erializer

r.KafkaProducer      : [Producer clientId=producer-1] Instantiated
s.AppInfoParser      : Kafka version: 3.6.0
s.AppInfoParser      : Kafka commitId: 60e845626d8a465a
s.AppInfoParser      : Kafka startTimeMs: 1701291442289
nts.Metadata         : [Producer clientId=producer-1] Cluster ID: 0
ransactionManager    : [Producer clientId=producer-1] ProducerId se
ctServiceImpl        : Partition: 2
ctServiceImpl        : Topic: product-created-events-topic
ctServiceImpl        : Offset: 0
ctServiceImpl        : ***** Returning product id
```

I see partition number where the product created event was stored in Kafka topic.

1.25x 2:32 / 2:45



But what if leader broker goes down before sharing this message with its followers and never comes back?



And in those cases, when it is critical to ensure that the message does not get lost, you will want

- **spring.kafka.producer.acks=all**
  Waits for an acknowledgement from all brokers.

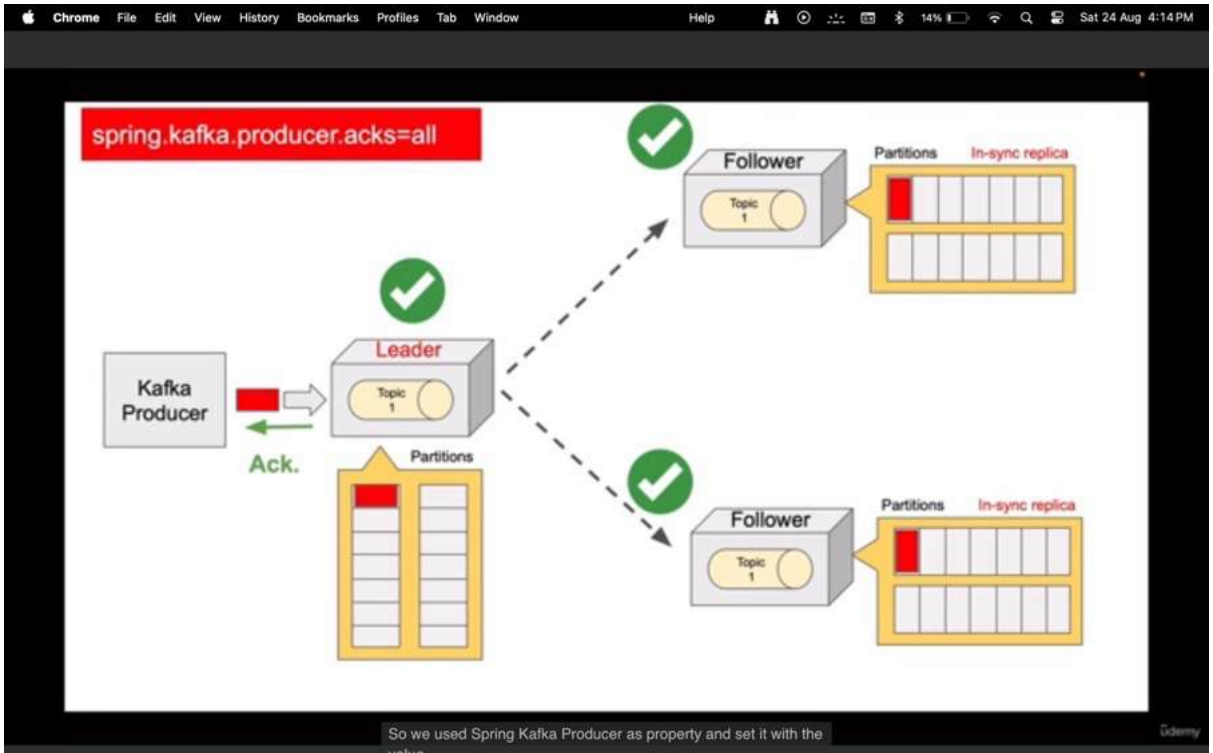
This option provides the strongest durability guarantees because no data will be lost as long as at

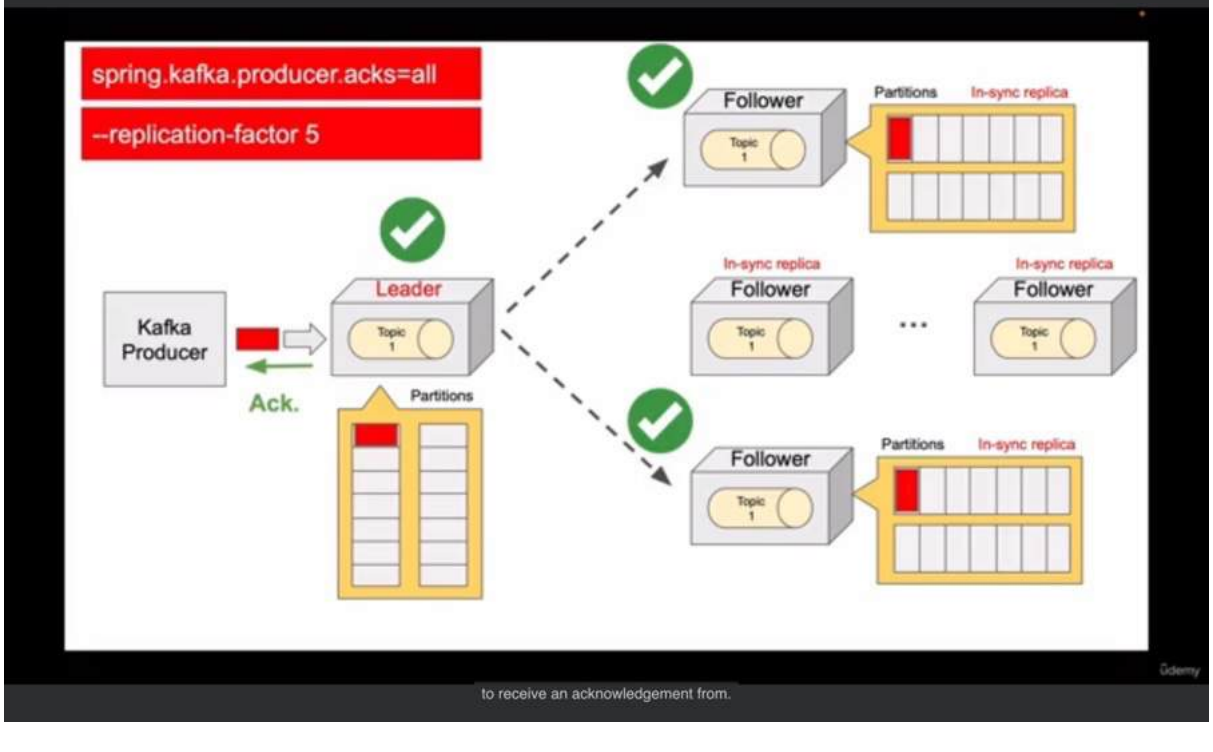
- **spring.kafka.producer.acks=all**
  Waits for an acknowledgement from all brokers.
- **spring.kafka.producer.acks=1**
  Waits for an acknowledgement from a leader broker only.

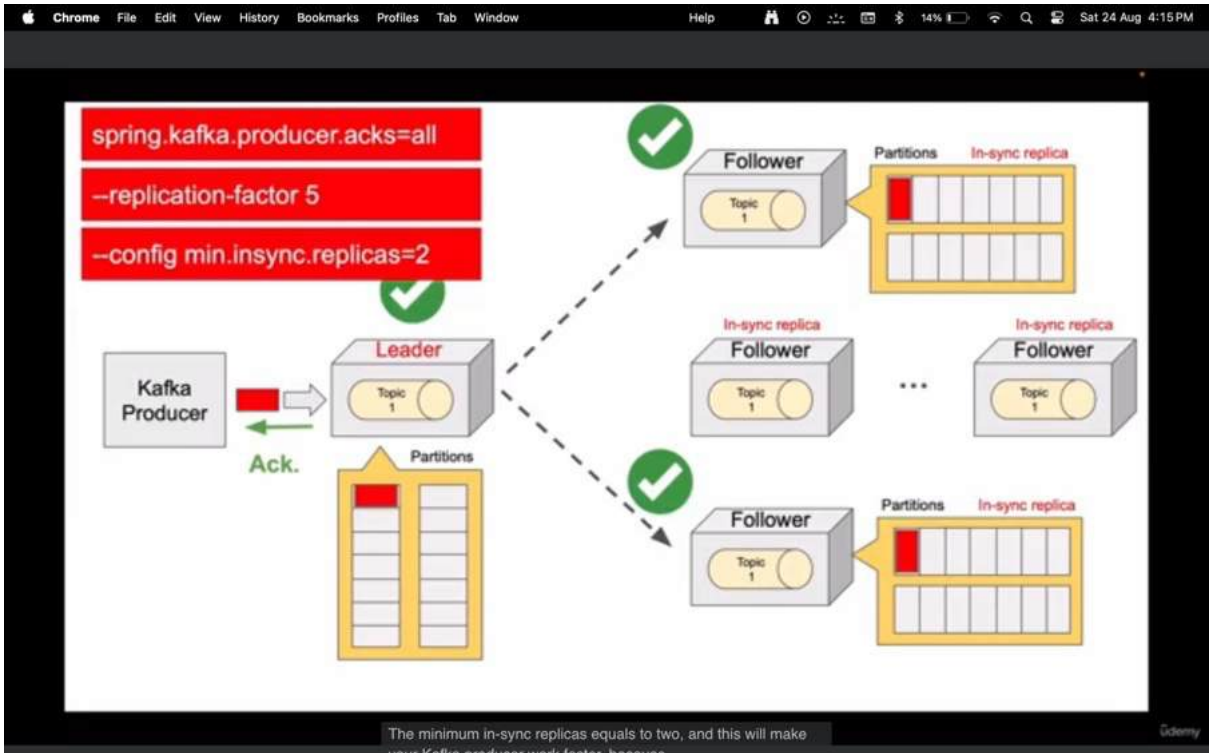
waiting for all brokers to acknowledge, but it is less reliable.

- **spring.kafka.producer.acks=all**
  Waits for an acknowledgement from all brokers.
- **spring.kafka.producer.acks=1**
  Waits for an acknowledgement from a leader broker only.
- **spring.kafka.producer.acks=0**
  Does not wait for an acknowledgement.

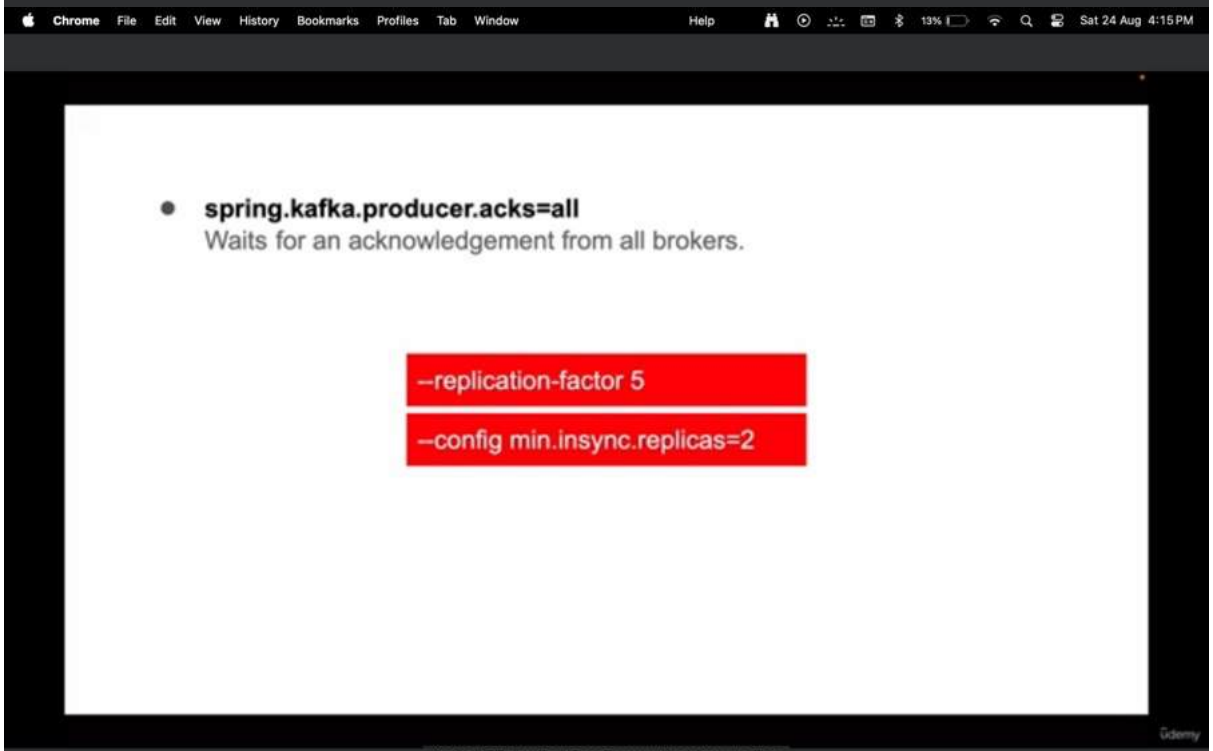
And because Kafka Producer does not wait, your system works faster, but there is no guarantee that


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sat 24 Aug 4:14 PM
spring.kafka.producer.acks=all
Kafka Producer
Ack.
Leader
Topic 1
Partitions
Follower
Partitions
In-sync replica
Follower
Partitions
In-sync replica
So we used Spring Kafka Producer as property and set it with the value.
Udemy

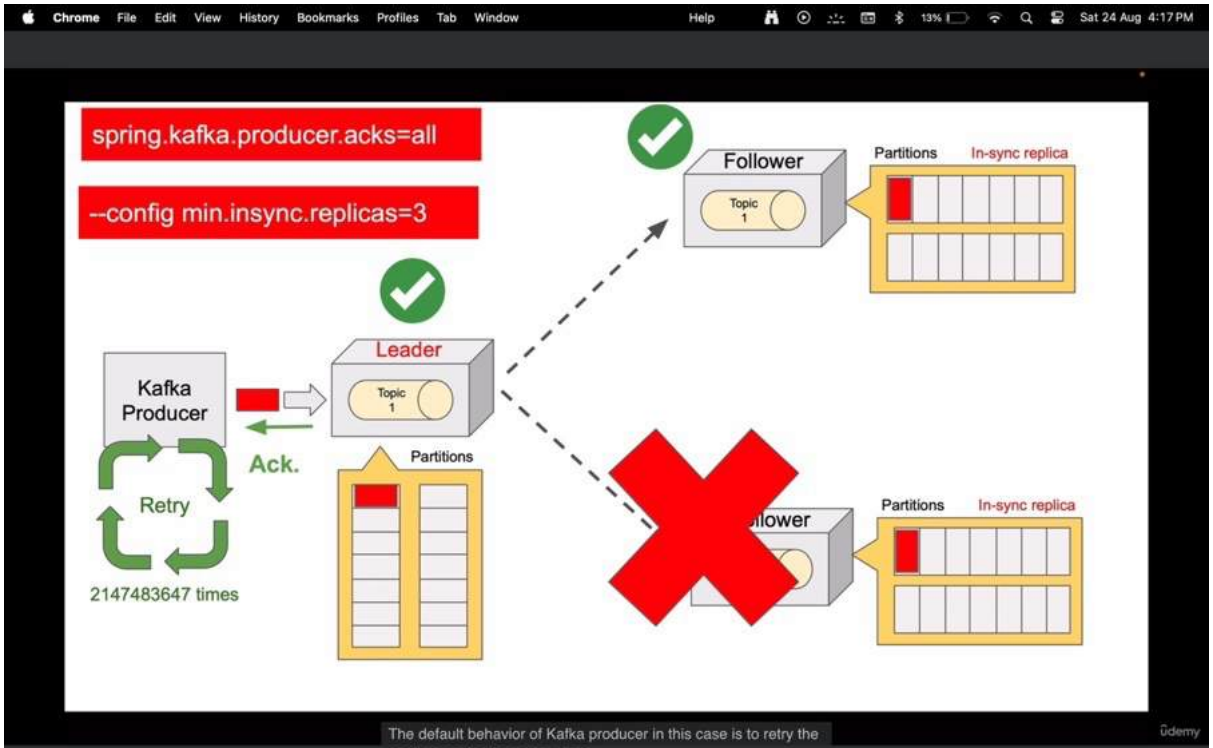

spring.kafka.producer.acks=all
--replication-factor 5
Kafka Producer
Ack.
Leader
Topic 1
Partitions
Follower
Partitions
In-sync replica
In-sync replica
Follower
In-sync replica
Follower
Follower
Partitions
In-sync replica
to receive an acknowledgement from.
Udemy



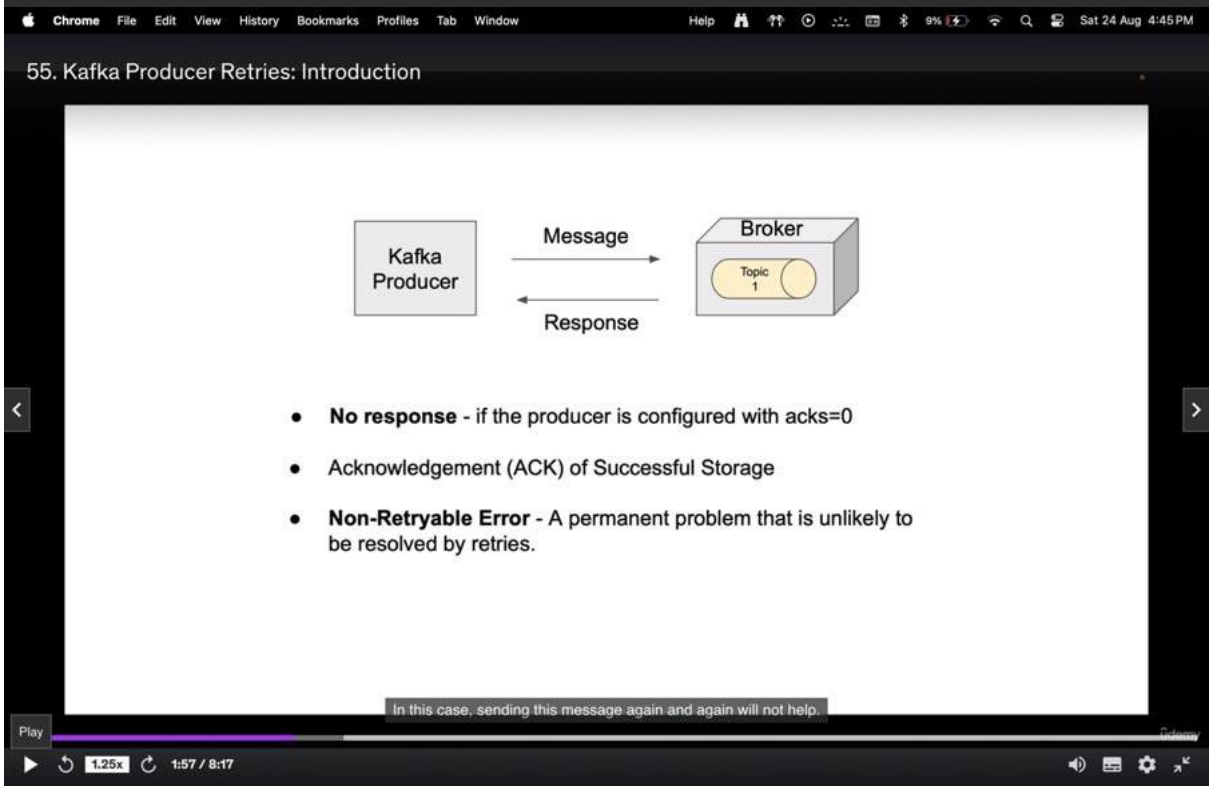
The minimum in-sync replicas equals to two, and this will make your Kafka producer work faster, because



And in the following lessons you will see how to do it.

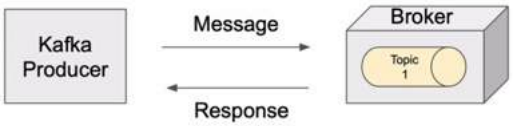


The default behavior of Kafka producer in this case is to retry the send operation for a very large



Chrome File Edit View History Bookmarks Profiles Tab Window Help 9% Sat 24 Aug 4:46 PM

55. Kafka Producer Retries: Introduction



- **No response** - if the producer is configured with acks=0
- Acknowledgement (ACK) of Successful Storage
- **Non-Retryable Error** - A permanent problem that is unlikely to be resolved by retries.
- **Retryable Error** - A temporary problem that can be resolved by retrying the send operation.

For example, if there is a network error or if the leader broker is being temporarily unavailable,

Pause
1.25x 2:11 / 8:17

Chrome File Edit View History Bookmarks Profiles Tab Window Help 13% Sat 24 Aug 4:47 PM

55. Kafka Producer Retries: Introduction

spring.kafka.producer.**retries**=10

How many times Kafka Producer will try to send a message before marking it as failed. Default value is 2147483647.

spring.kafka.producer.properties.**retry.backoff.ms**=1000

How long the producer will wait before attempting to retry a failed request. Default value is 100 ms.

it is a positive integer.

1.25x 3:59 / 8:17

Chrome File Edit View History Bookmarks Profiles Tab Window Help 13% Sat 24 Aug 4:48 PM

55. Kafka Producer Retries: Introduction

spring.kafka.producer.properties.**delivery.timeout.ms**=120000

The maximum time Producer can spend trying to deliver the message. Default value is 120000 ms (2 minutes).

Kafka producer can spend trying to send a message, and you can do it with this delivery timeout configuration

1.25x 4:47 / 8:17

Chrome File Edit View History Bookmarks Profiles Tab Window Help 22% Sat 24 Aug 4:52 PM

spring.kafka.producer.properties.**delivery.timeout.ms**=120000

The maximum time Producer can spend trying to deliver the message. Default value is 120000 ms (2 minutes).

delivery.timeout.ms >= linger.ms + request.timeout.ms

spring.kafka.producer.properties.**linger.ms**=0

The maximum time in milliseconds that the producer will wait and buffer data before sending a batch of messages. The default value is 0.

spring.kafka.producer.properties.**request.timeout.ms**=30000

The maximum time to wait for a response from the broker after sending a request. The default value is 30000 ms.

It is used to configure the maximum time producer will wait before sending a batch of messages to Kafka

Udemy

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializ
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonS
spring.kafka.producer.acks=all
```

And in the following lesson we will specify how many in-sync replicas should send this acknowledgement.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092
```

You have seen this command already in previous video lessons, because we used it to create a new topic.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092 --config min.insync.replicas=2
```

In this case the write operation will be successful if two replicas respond with acknowledgement.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092 --config min.insync.replicas=2
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: product-created-events-topic     TopicId: rl0tmMrLQ0enBdLvw2-EeQ PartitionCount: 3       ReplicationFactor: 3C
onfigs: min.insync.replicas=2,segment.bytes=1073741824
        Topic: product-created-events-topic     Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: product-created-events-topic     Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: product-created-events-topic     Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
Topic: topic2   TopicId: 67UVbERTSGUVjINbiZqXdg PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 1    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic2   Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

Here I have a topic with a name, product created events topic and it does already have configuration

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092 --config min.insync.replicas=2
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: product-created-events-topic     TopicId: rl0tmMrLQ0enBdLvw2-EeQ PartitionCount: 3       ReplicationFactor: 3C
onfigs: min.insync.replicas=2,segment.bytes=1073741824
        Topic: product-created-events-topic     Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: product-created-events-topic     Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: product-created-events-topic     Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
Topic: topic2   TopicId: 67UVbtRTSGOvjINbIzqXag PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 1    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic2   Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-configs.sh
```

So the new script is called Kafka Configs and it is used to add or update existing configuration.

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092 --config min.insync.replicas=2
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: product-created-events-topic     TopicId: rl0tmMrLQ0enBdLvw2-EeQ PartitionCount: 3       ReplicationFactor: 3C
onfigs: min.insync.replicas=2,segment.bytes=1073741824
        Topic: product-created-events-topic     Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: product-created-events-topic     Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: product-created-events-topic     Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
Topic: topic2   TopicId: 67UVbtRTSGOvjINbIzqXag PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 1    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic2   Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --entit
y-type topics --entity-name topic2 --add-config min.insync.replicas=2
```

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % pwd
/Users/sergeykargopolov/kafka
sergeykargopolov@Sergeys-MacBook-Pro kafka % /bin/kafka-topics.sh --create --topic insync-topic --partitions 3 --repl
ication-factor 3 --bootstrap-server localhost:9092 --config min.insync.replicas=2
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-topics.sh --describe --bootstrap-server localhost:9092
Topic: product-created-events-topic     TopicId: rl0tmMrLQ0enBdLvw2-EeQ PartitionCount: 3       ReplicationFactor: 3C
onfigs: min.insync.replicas=2,segment.bytes=1073741824
        Topic: product-created-events-topic     Partition: 0    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
        Topic: product-created-events-topic     Partition: 1    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: product-created-events-topic     Partition: 2    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
Topic: topic2   TopicId: 67UVbtRTSGOvjINbIzqXag PartitionCount: 3       ReplicationFactor: 3    Configs: segment.byte
s=1073741824
        Topic: topic2   Partition: 0    Leader: 2       Replicas: 2,3,1 Isr: 2,3,1
        Topic: topic2   Partition: 1    Leader: 3       Replicas: 3,1,2 Isr: 3,1,2
        Topic: topic2   Partition: 2    Leader: 1       Replicas: 1,2,3 Isr: 1,2,3
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-configs.sh --bootstrap-server localhost:9092 --alter --entit
y-type topics --entity-name topic2 --add-config min.insync.replicas=2
Completed updating config for topic topic2.
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

Our configuration should now be updated to see if it was applied to my topic.

58. Trying how the min.insync.replicas works

```
er.ReplicaFetcherManager)
[2023-12-01 09:29:41,550] INFO [ReplicaFetcher replicaId=3, d=2, fetcherId=0] Truncating partition topic2-0 wi
th TruncationState(offset=0, completed=true) due to local hi rmark 0 (kafka.server.ReplicaFetcherThread)
[2023-12-01 09:29:41,550] INFO [UnifiedLog partition=topic dir=/tmp/server-3/kraft-combined-logs] Truncating to 0
 has no effect as the largest offset in the log is -1 (k log.UnifiedLog)
[2023-12-01 09:29:41,579] INFO [ReplicaFetcher replica leaderId=1, fetcherId=0] Truncating partition topic2-2 wi
th TruncationState(offset=0, completed=true) due to l high watermark 0 (kafka.server.ReplicaFetcherThread)
[2023-12-01 09:29:41,588] INFO [UnifiedLog partitio ic2-2, dir=/tmp/server-3/kraft-combined-logs] Truncating to 0
 has no effect as the largest offset in the log i kafka.log.UnifiedLog)
[2023-12-01 09:39:45,160] INFO [BrokerToControll nnelManager id=3 name=forwarding] Node 1 disconnected. (org.apac
he.kafka.clients.NetworkClient)
[2023-12-01 10:02:42,490] INFO [DynamicConfigPublisher broker id=3] Updating topic topic2 with new configuration : mi
n.insync.replicas -> 2 (kafka.server.metadata.DynamicConfigPublisher)
log4j:ERROR Failed to rename [/Users/sergeykargopolov/kafka/bin/../logs/server.log] to [/Users/sergeykargopolov/kafka
/bin/../logs/server.log.2023-12-01-09].
[2023-12-01 10:13:12,883] INFO [BrokerToControllerChannelManager id=3 name=forwarding] Node 1 disconnected. (org.apac
he.kafka.clients.NetworkClient)
[2023-12-01 10:16:38,430] INFO [SnapshotGenerator id=3] Creating new KRaft snapshot file snapshot 000000000000000719
9-0000000004 because we have waited at least 60 minute(s). (org.apache.kafka.image.publisher.SnapshotGenerator)
[2023-12-01 10:16:38,531] INFO [SnapshotEmitter id=3] Successfully wrote snapshot 000000000000000007199-0000000004 (or
g.apache.kafka.image.publisher.SnapshotEmitter)
[2023-12-01 11:16:38,970] INFO [SnapshotGenerator id=3] Creating new KRaft snapshot file snapshot 000000000000001433
1-0000000004 because we have waited at least 60 minute(s). (org.apache.kafka.image.publisher.SnapshotGenerator)
[2023-12-01 11:16:39,017] INFO [SnapshotEmitter id=3] Successfully wrote snapshot 000000000000000014331-0000000004 (or
g.apache.kafka.image.publisher.SnapshotEmitter)
[2023-12-01 12:16:39,314] INFO [SnapshotGenerator id=3] Creating new KRaft snapshot file snapshot 000000000000002148
6-0000000004 because we have waited at least 60 minute(s). (org.apache.kafka.image.publisher.SnapshotGenerator)
[2023-12-01 12:16:39,363] INFO [SnapshotEmitter id=3] Successfully wrote snapshot 000000000000000021486-0000000004 (or
g.apache.kafka.image.publisher.SnapshotEm
```

And I have server three running in a separate tab.

1.25x 0:48 / 5:52

58. Trying how the min.insync.replicas works

```java
@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("topic2",productId, productCreatedEvent).get();

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());
        LOGGER.info("Offset: " + result.getRecordMetadata().offset());

        LOGGER.info("***** Returning product id");

        return productId;
    }
```

Just so that I can demonstrate acknowledgement feature.

1.25x 1:44 / 5:52

58. Trying how the min.insync.replicas works

```java
@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger LOGGER  = LoggerFactory.getLogger(this.getClass());

    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event
```

```
afka.common.utils.AppInfoParser     : Kafka version: 3.6.0
afka.common.utils.AppInfoParser     : Kafka commitId: 60e845626d8a465a
afka.common.utils.AppInfoParser     : Kafka startTimeMs: 1701454723472
afka.common.utils.AppInfoParser     : App info kafka.admin.client for adminclient-1 unregistered
ache.kafka.common.metrics.Metrics   : Metrics scheduler closed
ache.kafka.common.metrics.Metrics   : Closing reporter org.apache.kafka.common.metrics.JmxReporte
ache.kafka.common.metrics.Metrics   : Metrics reporters closed
                                    : Started ProductsMicroserviceApplication in 2.192 seconds (p
```

My microservice is up and running, I will expand console and port [illegible] (http) with context path ''
we'll copy the port number on which this

1.25x 2:17 / 5:52

58. Trying how the min.insync.replicas works

POST http://localhost:50191

http://localhost:50191/products

POST http://localhost:50191/products

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Body Cookies Headers (5) Test Results

Status: 201 Created Time: 442 ms Size: 205 B

Pretty Raw Preview Visualize Text

```
178e2614-e7c5-4fc7-bac9-1af3651d6820
```

The request is successful.

1.25x 2:45 / 5:52

58. Trying how the min.insync.replicas works

```
Last login: Fri Dec  1 09:16:50 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic topic2 --bootstrap-server localh
ost:9092
{"productId":"178e2614-e7c5-4fc7-bac9-1af3651d6820","title":"iPhone 11","price":800,"quantity":19}
```

Now let's stop one of the servers.

1.25x 2:56 / 5:52

58. Trying how the min.insync.replicas works

```
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-Produce]: Shutdown completed (kafka.server.Client
QuotaManager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-Request]: Shutting down (kafka.server.ClientQuota
Manager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-Request]: Stopped (kafka.server.ClientQuotaManage
r$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-Request]: Shutdown completed (kafka.server.Client
QuotaManager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-ControllerMutation]: Shutting down (kafka.server.
ClientQuotaManager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-ControllerMutation]: Stopped (kafka.server.Client
QuotaManager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [controller-3-ThrottledChannelReaper-ControllerMutation]: Shutdown completed (kafka.se
rver.ClientQuotaManager$ThrottledChannelReaper)
[2023-12-01 13:20:03,380] INFO [QuorumController id=3] closed event queue. (org.apache.kafka.queue.KafkaEventQueue)
[2023-12-01 13:20:03,382] INFO [SharedServer id=3] Stopping SharedServer (kafka.server.SharedServer)
[2023-12-01 13:20:03,383] INFO [MetadataLoader id=3] beginShutdown: shutting down event queue. (org.apache.kafka.queu
e.KafkaEventQueue)
[2023-12-01 13:20:03,383] INFO [SnapshotGenerator id=3] close: shutting down event queue. (org.apache.kafka.queue.Kaf
kaEventQueue)
[2023-12-01 13:20:03,383] INFO [SnapshotGenerator id=3] closed event queue. (org.apache.kafka.queue.KafkaEventQueue)
[2023-12-01 13:20:03,383] INFO [MetadataLoader id=3] closed event queue. (org.apache.kafka.queue.KafkaEventQueue)
[2023-12-01 13:20:03,384] INFO [SnapshotGenerator id=3] closed event queue. (org.apache.kafka.queue.KafkaEventQueue)
[2023-12-01 13:20:03,385] INFO Metrics scheduler closed (org.apache.kafka.common.metrics.Metrics)
[2023-12-01 13:20:03,385] INFO Closing reporter org.apache.kafka.common.metrics.JmxReporter (org.apache.kafka.common.
metrics.Metrics)
[2023-12-01 13:20:03,386] INFO Metrics reporters closed (org.apache.kafka.common.metrics.Metrics)
[2023-12-01 13:20:03,386] INFO App info kafka.server for 3 unregistered (org.apache.kafka.common.utils.AppInfoParser)
[2023-12-01 13:20:03,386] INFO App info kafka.server.KafkaServer for 3 unregistered (org.apache.kafka.common.utils.AppInfoParser)
sergeykargopolov@Sergeys-MacBook-Pro kafka %
```

Now I have only two servers running

1.25x 3:05 / 5:52

```
POST http://localhost:50191/products
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

```
Status: 201 Created  Time: 53 ms  Size: 205 B
8e2bc122-dd1f-49c7-b434-f424a4ba641e
```

Let's see if Kafka Consumer was able to consume a new message.

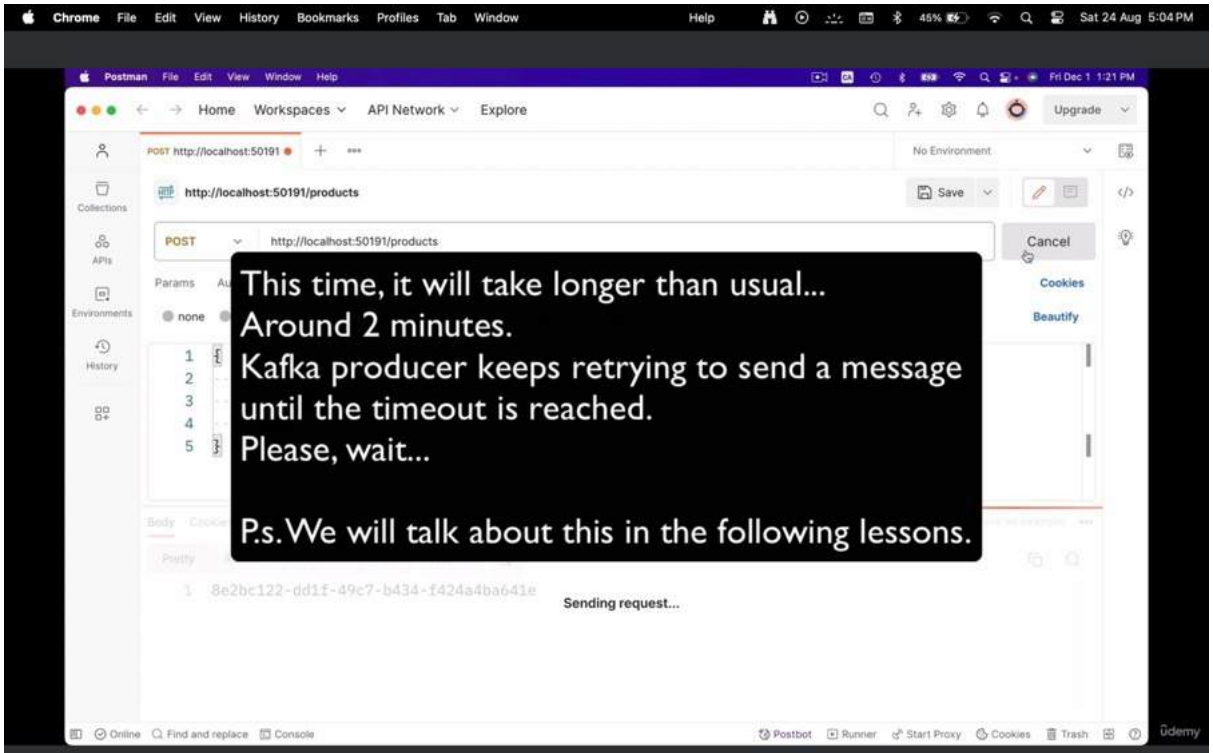
```
Last login: Fri Dec  1 09:16:50 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic topic2 --bootstrap-server localh
ost:9092
{"productId":"178e2614-e7c5-4fc7-bac9-1af3651d6820","title":"iPhone 11","price":800,"quantity":19}
{"productId":"8e2bc122-dd1f-49c7-b434-f424a4ba641e","title":"iPhone 11","price":800,"quantity":19}
```

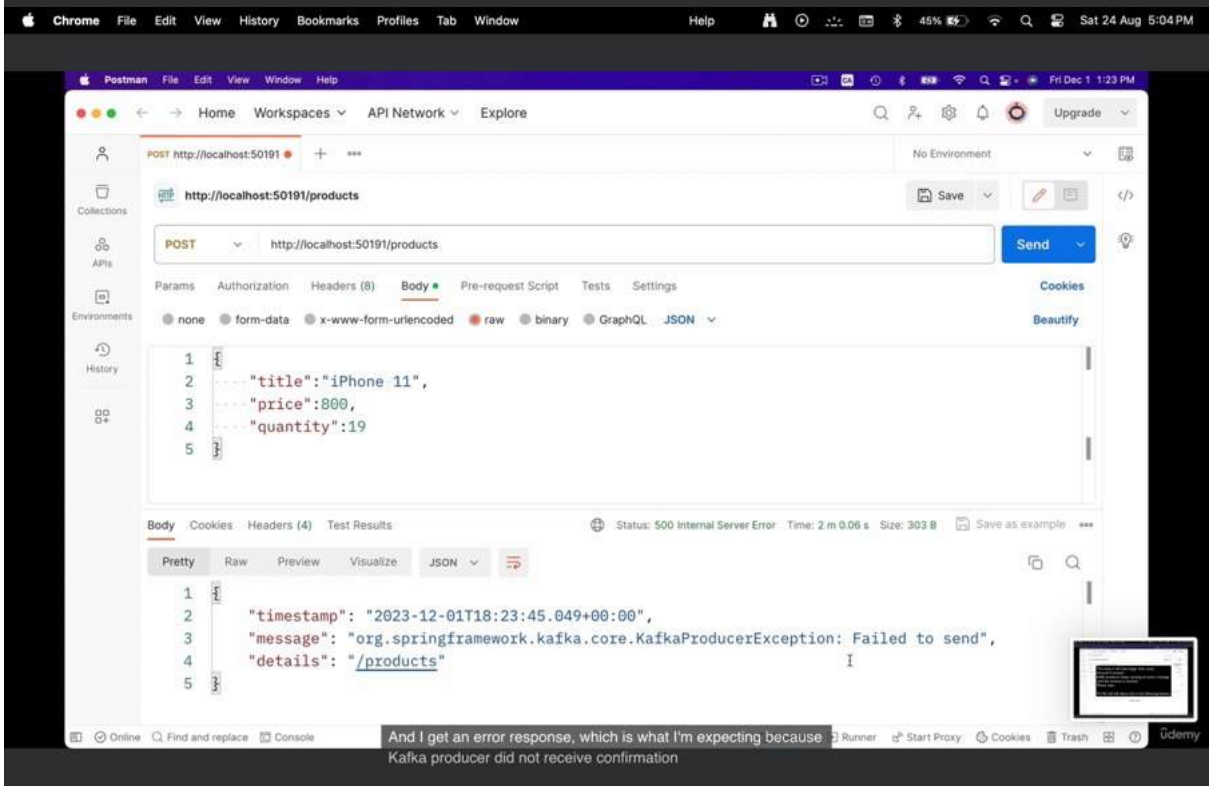
So it worked.

```
e from node 3 (org.apache.kafka.clients.NetworkClient)
[2023-12-01 13:20:00,148] INFO [ReplicaFetcher replicaId=2, leaderId=3, fetcherId=0] Cancelled in-flight FETCH reques
t with correlation id 28920 due to node 3 being disconnected (elapsed time since creation: 460ms, elapsed time since
send: 460ms, request timeout: 30000ms) (org.apache.kafka.clients.NetworkClient)
[2023-12-01 13:20:00,150] INFO [ReplicaFetcher replicaId=2, leaderId=3, fetcherId=0] Error sending fetch request (ses
sionId=1245954787, epoch=28920) to node 3: (org.apache.kafka.clients.FetchSessionHandler)
java.io.IOException: Client was shutdown before response was read
        at org.apache.kafka.clients.NetworkClientUtils.sendAndReceive(NetworkClientUtils.java:108)
        at kafka.server.BrokerBlockingSender.sendRequest(BrokerBlockingSender.scala:113)
        at kafka.server.RemoteLeaderEndPoint.fetch(RemoteLeaderEndPoint.scala:79)
        at kafka.server.AbstractFetcherThread.processFetchRequest(AbstractFetcherThread.scala:316)
        at kafka.server.AbstractFetcherThread.$anonfun$maybeFetch$3(AbstractFetcherThread.scala:130)
        at kafka.server.AbstractFetcherThread                                   d(AbstractFetcherThread.scala:129)
        at scala.Option.foreach(Option.scala:
        at kafka.server.AbstractFetcherThread                                   read.scala:129)
        at kafka.server.AbstractFetcherThread.doWork(AbstractFetcherThread.scala:112)
        at kafka.server.ReplicaFetcherThread.doWork(ReplicaFetcherThread.scala:98)
        at org.apache.kafka.server.util.ShutdownableThread.run(ShutdownableThread.java:130)
[2023-12-01 13:20:00,156] INFO [ReplicaFetcherThread-0-3]: Stopped (kafka.server.ReplicaFetcherThread)
[2023-12-01 13:20:00,157] INFO [ReplicaFetcherThread-0-3]: Shutdown completed (kafka.server.ReplicaFetcherThread)
[2023-12-01 13:20:00,523] INFO [ReplicaFetcher replicaId=2, leaderId=1, fetcherId=0] Truncating partition topic2-1 wi
th TruncationState(offset=0, completed=true) due to local high watermark 0 (kafka.server.ReplicaFetcherThread)
[2023-12-01 13:20:00,523] INFO [UnifiedLog partition=topic2-1, dir=/tmp/server-2/kraft-combined-logs] Truncating to 0
 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
[2023-12-01 13:20:00,523] INFO [ReplicaFetcher replicaId=2, leaderId=1, fetcherId=0] Truncating partition product-cre
ated-events-topic-2 with TruncationState(offset=0, completed=true) due to local high watermark 0 (kafka.server.Re
aFetcherThread)
[2023-12-01 13:20:00,523] INFO [UnifiedLog partition=product-created-events-topic-2, dir=/tmp/server-2/kraft-comb
-logs] Truncating to 0 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
```

CONTROL + C

Now let's stop one more server I will select server two and I will stop this server.


POST http://localhost:50191/products
Cancel
This time, it will take longer than usual...
Around 2 minutes.
Kafka producer keeps retrying to send a message
until the timeout is reached.
Please, wait...
P.s. We will talk about this in the following lessons.
8e2bc122-dd1f-49c7-b434-f424a4ba641e
Sending request...


POST http://localhost:50191/products
Send
"title":"iPhone 11",
"price":800,
"quantity":19
Status: 500 Internal Server Error
"timestamp": "2023-12-01T18:23:45.049+00:00",
"message": "org.springframework.kafka.core.KafkaProducerException: Failed to send",
"details": "/products"
And I get an error response, which is what I'm expecting because
Kafka producer did not receive confirmation

```
Last login: Fri Dec  1 09:16:50 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --topic topic2 --bootstrap-server localh
ost:9092
{"productId":"178e2614-e7c5-4fc7-bac9-1af3651d6820","title":"iPhone 11","price":800,"quantity":19}
{"productId":"8e2bc122-dd1f-49c7-b434-f424a4ba641e","title":"iPhone 11","price":800,"quantity":19}
```

Select Kafka consumer and there is no new message.

```
on topic-partition topic2-1, retrying (2147482486 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482485 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482484 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482483 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482482 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482481 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482480 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482479 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482478 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482477 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482472 attempts left). Error: NOT_ENOUGH_REPLICAS
on topic-partition topic2-1, retrying (2147482471 attempts left). Error: NOT_ENOUGH_REPLICAS
dff' and payload='com.appsdeveloperblog.ws.products.service.ProductCreatedEvent@2143f8ca' to topic
```

Error: NOT_ENOUGH_REPLICAS

Not enough replicas which will eventually cause Kafka producer exception.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 47% Sat 24 Aug 5:05 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Dec 1 1:25 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

Package Explorer
- ProductsMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.products
    - com.appsdeveloperblog.ws.products.rest
    - com.appsdeveloperblog.ws.products.service
      - ProductCreatedEvent.java
      - ProductService.java
      - ProductServiceImpl.java
  - src/main/resources
    - static
    - templates
    - application.properties
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

Boot Dashboard
- local
  - ProductsMicroservice [:50191]

application.properties | ProductServiceImpl.java

```
13
14 @Service
```

Problems | Javadoc | Declaration | Console | Progress

```
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
WARN 23786 --- [ad | producer-1] o.a.k.clients.producer.internals.Sender  : [Producer clientId=p
ERROR 23786 --- [ad | producer-1] o.s.k.support.LoggingProducerListener    : Exception thrown whe

s.TimeoutException: Expiring 1 record(s) for topic2-1:120001 ms has passed since batch creation

ERROR 23786 --- [o-auto-1-exec-8] c.a.ws.products.rest.ProductController   : org.springframework.

nException: org.springframework.kafka.core.KafkaProducerException: Failed to send
l.concurrent.CompletableFuture.reportGet(CompletableFuture.java:396) ~[na:na]
l.concurrent.CompletableFuture.get(CompletableFuture.java:2073) ~[na:na]
log.ws.products.service.ProductServiceImpl.createProduct(ProductServiceImpl.java:38) ~[classes/:na
log.ws.products.rest.ProductController.createProduct(ProductController.java:32) ~[classes/:na]
rnal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:na]
rnal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) ~[na:na]
rnal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:na]
g.reflect.Method.invoke(Method.java:568) ~[na:na]
k.web.method.support.InvocableHandlerMethod.doInvoke(InvocableHandlerMethod.java:254) ~[spring-web
k.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:182) ~[sp
k.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocable
k.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappin
k.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHand
k.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)
k.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1089) ~[spring-webmvc-6.1.1.jar:
k.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:979) ~[spring-webmvc-6.1.1.jar:6.
k.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1014) ~[spring-webmvc-6.1.1.ja
k.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:914) ~[spring-webmvc-6.1.1.jar:6.1.1]
```

This is the one Kafka producer exception with a message failed to send.

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 48% Sat 24 Aug 5:07 PM

SpringToolSuite4 File Edit Source Navigate Search Project Run Window Help Fri Dec 1 3:20 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

Package Explorer
- ProductsMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.products
    - com.appsdeveloperblog.ws.products.rest
    - com.appsdeveloperblog.ws.products.service
  - src/main/resources
    - static
    - templates
    - application.properties
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml

Boot Dashboard
- local
  - ProductsMicroservice

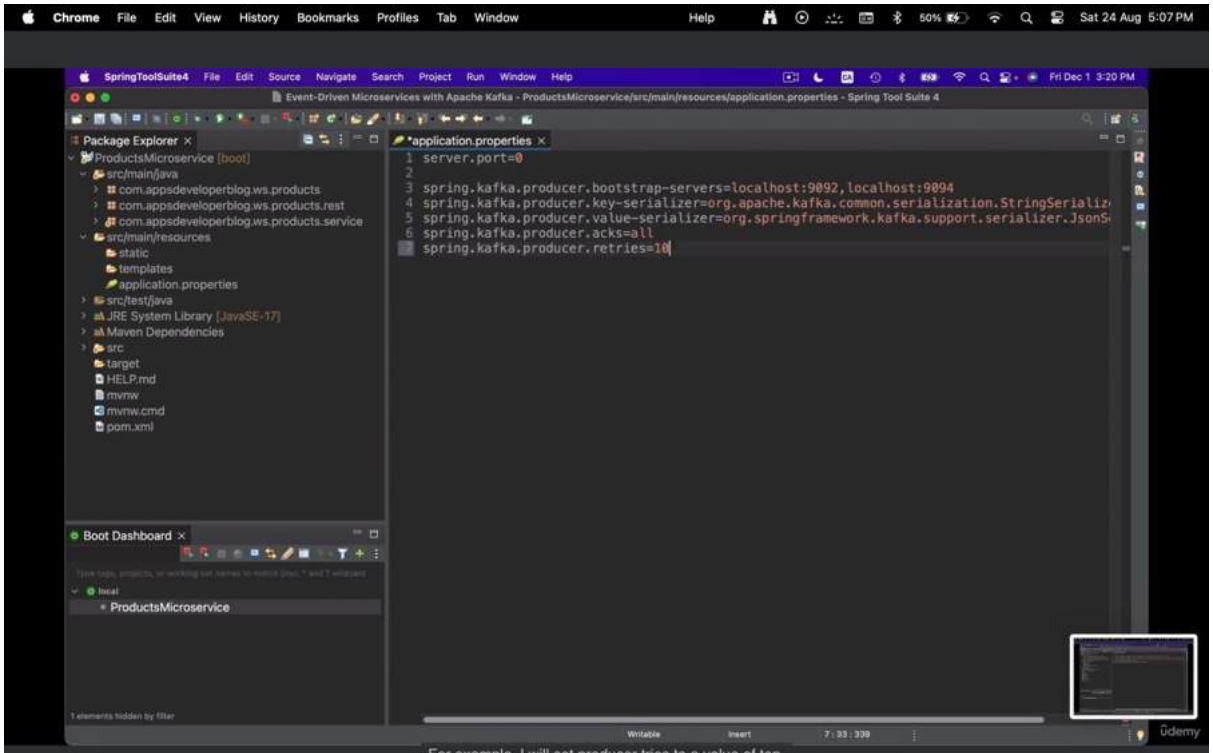
*application.properties

```
1 server.port=0
2
3 spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
4 spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializ
5 spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonS
6 spring.kafka.producer.acks=all
7 spring.kafka.producer.retries=2147483647
```

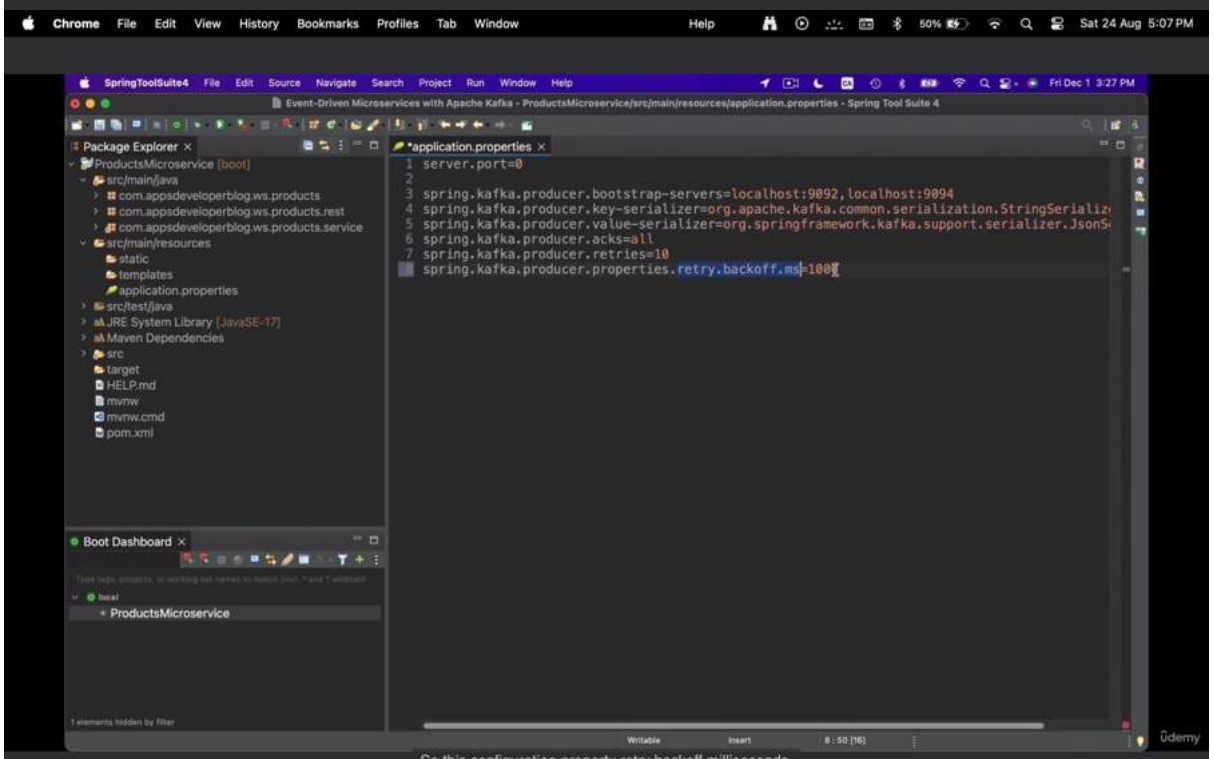
Writable Insert 7 : 41 : 347

Udemy

So for us to see how it works, it is better to set a smaller value.



For example, I will set producer tries to a value of ten.



So this configuration property retry backoff milliseconds.

60. Trying how Kafka Producer Retries work

```
[2023-12-02 10:02:14,441] WARN [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition topic2-0 marked as failed (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,441] INFO [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition product-created-events-topic-1 has an older epoch (95) than the current leader. Will await the new LeaderAndIsr state before resuming fetching. (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,441] WARN [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition product-created-events-topic-1 marked as failed (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,441] INFO [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition product-created-events-topic-2 has an older epoch (91) than the current leader. Will await the new LeaderAndIsr state before resuming fetching. (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,441] WARN [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition product-created-events-topic-2 marked as failed (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,748] INFO [ReplicaFetcherManager on broker 3] Removed fetcher for partitions Set(topic2-1, product-created-events-topic-2) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,820] INFO [ReplicaFetcherManager on broker 3] Removed fetcher for partitions Set(topic2-0, product-created-events-topic-1) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,824] INFO [ReplicaFetcherThread-0-2]: Starting (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,824] INFO [ReplicaFetcherManager on broker 3] Added fetcher to broker 2 for partitions HashMap(topic2-0 -> InitialFetchState(Some(67UVbtRTSGOvjINbIzqXag),BrokerEndPoint(id=2, host=localhost:9094),96,1), product-created-events-topic-1 -> InitialFetchState(Some(rl0tmMrLQ0enBdLvw2-EeQ),BrokerEndPoint(id=2, host=localhost:9094),96,0)) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,841] INFO [ReplicaFetcher replicaId=3, leaderId=2, fetcherId=0] Truncating partition product-created-events-topic-1 with TruncationState(offset=0, completed=true) due to local high watermark 0 (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,841] INFO [UnifiedLog partition=product-created-events-topic-1, dir=/tmp/server-3/kraft-combined-logs] Truncating to 0 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
[2023-12-02 10:08:05,460] INFO [RaftManager id=3] Node 2 disconnected. (org.apache.kafka.clients.NetworkClient)
[2023-12-02 10:09:12,689] INFO [RaftManager id=3] Node 1 disconnected. (org.apache.kafka.clients.NetworkClient)
```

It is also running and this is a terminal window of Kafka Broker three.

60. Trying how Kafka Producer Retries work

POST http://localhost:55722/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

```
dcb163d9-4669-4216-9064-6a17345c5be7
```

Now I'm going to stop one Kafka broker.

60. Trying how Kafka Producer Retries work

```
 fetching. (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,441] WARN [ReplicaFetcher replicaId=3, leaderId=1, fetcherId=0] Partition product-created-eve
nts-topic-2 marked as failed (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,748] INFO [ReplicaFetcherManager on broker 3] Removed fetcher for partitions Set(topic2-1, pr
oduct-created-events-topic-2) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,820] INFO [ReplicaFetcherManager on broker 3] Removed fetcher for partitions Set(topic2-0, pr
oduct-created-events-topic-1) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,824] INFO [ReplicaFetcherThread-0-2]: Starting (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:02:14,824] INFO [ReplicaFetcherManager on broker 3] Added fetcher to broker 2 for partitions HashMa
p(topic2-0 -> InitialFetchState(Some(67UVbtRTSGOvjINbIzqXag),BrokerEndPoint(id=2, host=localhost:9094),96,1), prod
uct-created-events-topic-1 -> InitialFetchState(Some(rl0tmMrLQ0enBdLvw2-EeQ),BrokerEndPoint(id=2, host=localhost:9
094),96,0)) (kafka.server.ReplicaFetcherManager)
[2023-12-02 10:02:14,841] INFO [ReplicaFetcher                                  2, fetcherId=0] Truncating partition product-
created-events-topic-1 with TruncationState(off                                  due to local high watermark 0 (kafka.server.
ReplicaFetcherThread)
[2023-12-02 10:02:14,841] INFO [UnifiedLog partition=product-created-events-topic-1, dir=/tmp/server-3/kraft-combi
ned-logs] Truncating to 0 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
[2023-12-02 10:08:05,460] INFO [RaftManager id=3] Node 2 disconnected. (org.apache.kafka.clients.NetworkClient)
[2023-12-02 10:09:12,689] INFO [BrokerToControllerChannelManager id=3 name=forwarding] Node 1 disconnected. (org.a
pache.kafka.clients.NetworkClient)
log4j:ERROR Failed to rename [/Users/sergeykargopolov/kafka/bin/../logs/controller.log] to [/Users/sergeykargopolo
v/kafka/bin/../logs/controller.log.2023-12-02-09].
^C[2023-12-02 10:19:23,945] INFO Terminating process due to signal SIGINT (org.apache.kafka.common.utils.LoggingSi
gnalHandler)
[2023-12-02 10:19:23,946] INFO [BrokerServer id=3] Transition from STARTED to SHUTTING_DOWN (kafka.server.BrokerSe
rver)
[2023-12-02 10:19:23,946] INFO [BrokerServer id=3] shutting down (kafka.server.BrokerServer)
[2023-12-02 10:19:23,947] INFO [BrokerLifecycleManager id=3] Beginning controlled shutdown. (kafka.server.Brok
fecycleManager)
```

Control + C

1:42 / 3:52

60. Trying how Kafka Producer Retries work

```
lose from node 3 (org.apache.kafka.clients.NetworkClient)
[2023-12-02 10:19:24,403] INFO [ReplicaFetcher replicaId=2, leaderId=3, fetcherId=0] Cancelled in-flight FETCH req
uest with correlation id 2050 due to node 3 being disconnected (elapsed time since creation: 392ms, elapsed time s
ince send: 392ms, request timeout: 30000ms) (org.apache.kafka.clients.NetworkClient)
[2023-12-02 10:19:24,405] INFO [ReplicaFetcher replicaId=2, leaderId=3, fetcherId=0] Error sending fetch request (
sessionId=2005616801, epoch=2050) to node 3: (org.apache.kafka.clients.FetchSessionHandler)
java.io.IOException: Client was shutdown before response was read
        at org.apache.kafka.clients.NetworkClientUtils.sendAndReceive(NetworkClientUtils.java:108)
        at kafka.server.BrokerBlockingSender.sendRequest(BrokerBlockingSender.scala:113)
        at kafka.server.RemoteLeaderEndPoint.fetch(RemoteLeaderEndPoint.scala:79)
        at kafka.server.AbstractFetcherThread.processFetchRequest(AbstractFetcherThread.scala:316)
        at kafka.server.AbstractFetcherThread.$anonfun$maybeFetch$3(AbstractFetcherThread.scala:130)
        at kafka.server.AbstractFetcherThread.$                                  dapted(AbstractFetcherThread.scala:129)
        at scala.Option.foreach(Option.scala:43
        at kafka.server.AbstractFetcherThread.m                                  herThread.scala:129)
        at kafka.server.AbstractFetcherThread.doWork(AbstractFetcherThread.scala:112)
        at kafka.server.ReplicaFetcherThread.doWork(ReplicaFetcherThread.scala:98)
        at org.apache.kafka.server.util.ShutdownableThread.run(ShutdownableThread.java:130)
[2023-12-02 10:19:24,407] INFO [ReplicaFetcherThread-0-3]: Stopped (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:19:24,407] INFO [ReplicaFetcherThread-0-3]: Shutdown completed (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:19:24,641] INFO [ReplicaFetcher replicaId=2, leaderId=1, fetcherId=0] Truncating partition topic2-1
 with TruncationState(offset=0, completed=true) due to local high watermark 0 (kafka.server.ReplicaFetcherThread)
[2023-12-02 10:19:24,642] INFO [UnifiedLog partition=topic2-1, dir=/tmp/server-2/kraft-combined-logs] Truncating t
o 0 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
[2023-12-02 10:19:24,642] INFO [ReplicaFetcher replicaId=2, leaderId=1, fetcherId=0] Truncating partition product-
created-events-topic-2 with TruncationState(offset=0, completed=true) due to local high watermark 0 (kafka.server.
ReplicaFetcherThread)
[2023-12-02 10:19:24,642] INFO [UnifiedLog partition=product-created-events-topic-2, dir=/tmp/server-2/kraft-combi
ned-logs] Truncating to 0 has no effect as the largest offset in the log is -1 (kafka.log.UnifiedLog)
```

Control + C

2:16 / 3:52

60. Trying how Kafka Producer Retries work

```
server.port=0
```

```
Problems  Javadoc  Declaration  Console  Progress

in idempotent producer.

-M75dt-RUqu9_fxKZH46w
: to 3000 with epoch 0

nected.
node 3 (localhost/127.0.0.1:9096) could not be established. Broker may not be available.

nected.
nnected.

duce response with correlation id 11 on topic-partition topic2-1, retrying (9 attempts left). Err
duce response with correlation id 12 on topic-partition topic2-1, retrying (8 attempts left). Err
duce response with correlation id 13 on topic-partition topic2-1, retrying (7 attempts left). Err
duce response with correlation id 14 on topic-partition topic2-1, retrying (6 attempts left). Err
duce response with correlation id 15 on topic-partition topic2-1, retrying (5 attempts left). Err
duce response with correlation id 16 on topic-partition topic2-1, retrying (4 attempts left). Err
duce response with correlation id 17 on topic-partition topic2-1, retrying (3 attempts left). Err
duce response with correlation id 18 on topic-partition topic2-1, retrying (2 attempts left). Err
```

I see that my Kafka producer is trying to send Kafka message every second.

60. Trying how Kafka Producer Retries work

```
server.port=0
```

```
Problems  Javadoc  Declaration  Console  Progress

dEvent
de 3 disconnected.
nnection to node 3 (localhost/127.0.0.1:9096) could not be established. Broker may not be availa

de 2 disconnected.
de -2 disconnected.
dEvent
t error produce response with correlation id 11 on topic-partition topic2-1, retrying (9 attempt
t error produce response with correlation id 12 on topic-partition topic2-1, retrying (8 attempt
t error produce response with correlation id 13 on topic-partition topic2-1, retrying (7 attempt
t error produce response with correlation id 14 on topic-partition topic2-1, retrying (6 attempt
t error produce response with correlation id 15 on topic-partition topic2-1, retrying (5 attempt
t error produce response with correlation id 16 on topic-partition topic2-1, retrying (4 attempt
t error produce response with correlation id 17 on topic-partition topic2-1, retrying (3 attempt
t error produce response with correlation id 18 on topic-partition topic2-1, retrying (2 attempt
t error produce response with correlation id 19 on topic-partition topic2-1, retrying (1 attempt
t error produce response with correlation id 20 on topic-partition topic2-1, retrying (0 attempt
essage with key='c90e2837-a8fd-49e8-9b35-e0321b9bf803' and payload='com.appsdeveloperblog.ws.pro

fkaProducerException: Failed to send
```

60. Trying how Kafka Producer Retries work

```
1:9096) could not be established. Broker may not be available.

ion id 11 on topic-partition topic2-1, retrying (9 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 12 on topic-partition topic2-1, retrying (8 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 13 on topic-partition topic2-1, retrying (7 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 14 on topic-partition topic2-1, retrying (6 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 15 on topic-partition topic2-1, retrying (5 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 16 on topic-partition topic2-1, retrying (4 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 17 on topic-partition topic2-1, retrying (3 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 18 on topic-partition topic2-1, retrying (2 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 19 on topic-partition topic2-1, retrying (1 attempts left). Error: NOT_ENOUGH_REPLICAS
ion id 20 on topic-partition topic2-1, retrying (0 attempts left). Error: NOT_ENOUGH_REPLICAS
35-e0321b9bf803' and payload='com.appsdeveloperblog.ws.products.service.ProductCreatedEvent@7803
```

It did try to send Kafka message exactly ten times with a time interval of one second, and once it

60. Trying how Kafka Producer Retries work

POST http://localhost:55722/products

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 500 Internal Server Error Time: 10.04 s Size: 303 B

```
{
    "timestamp": "2023-12-02T15:21:28.184+00:00",
    "message": "org.springframework.kafka.core.KafkaProducerException: Failed to send",
    "details": "/products"
}
```

61. Kafka Producer Delivery & Request Timeout

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSer
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
```

The default value of this configuration property is two minutes, but if you decide to change it, then

0:31 / 3:55

61. Kafka Producer Delivery & Request Timeout

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSer
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000

spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000
```

And we can run our application and try how it works.

3:52 / 3:55

62. Trying how Kafka Producer Delivery & Request Timeout works

```
[2023-12-06 17:56:10,949] INFO [GroupCoordinator 3]: Resigned as the group coordinator for partition 8 in epoch Option
alInt[23] (kafka.coordinator.group.GroupCoordinator)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Finished unloading __consumer_offsets-39 for coordina
tor epoch OptionalInt[23]. Removed 0 cached offsets and 0 cached groups. (kafka.coordinator.group.GroupMetadataManager
)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Scheduling unloading of offsets and group metadata fr
om __consumer_offsets-8 (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupCoordinator 3]: Resigned as the group coordinator for partition 38 in epoch Optio
nalInt[23] (kafka.coordinator.group.GroupCoordinator)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Scheduling unloading of offsets and group metadata fr
om __consumer_offsets-38 (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Finished unloading __consumer_offsets-8 for coordinat
or epoch OptionalInt[23]. Removed 0 cached offset                                  (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupCoordinator                                   oup coordinator for partition 3 in epoch Option
alInt[23] (kafka.coordinator.group.GroupCoordinat
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Scheduling unloading of offsets and group metadata fr
om __consumer_offsets-3 (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Finished unloading __consumer_offsets-38 for coordina
tor epoch OptionalInt[23]. Removed 0 cached offsets and 0 cached groups. (kafka.coordinator.group.GroupMetadataManager
)
[2023-12-06 17:56:10,949] INFO [GroupCoordinator 3]: Resigned as the group coordinator for partition 33 in epoch Optio
nalInt[23] (kafka.coordinator.group.GroupCoordinator)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Finished unloading __consumer_offsets-3 for coordinat
or epoch OptionalInt[23]. Removed 0 cached offsets and 0 cached groups. (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Scheduling unloading of offsets and group metadata fr
om __consumer_offsets-33 (kafka.coordinator.group.GroupMetadataManager)
[2023-12-06 17:56:10,949] INFO [GroupMetadataManager brokerId=3] Finished unloading __consumer_offsets-33 for coordina
tor epoch OptionalInt[23]. Removed 0 cached offsets and 0 cached groups. (kafka.coordinator.group.GroupMetadataManager
)
```

Control + C

1.25x 1:40 / 3:07

62. Trying how Kafka Producer Delivery & Request Timeout works

```
aft.QuorumState)
[2023-12-06 17:56:14,352] INFO [RaftManager id=2] Completed transition to Voted(epoch=148, votedId=1, voters=[1, 2, 3]
, electionTimeoutMs=1334) from Unattached(epoch=148, voters=[1, 2, 3], electionTimeoutMs=1015) (org.apache.kafka.raft.
QuorumState)
[2023-12-06 17:56:14,352] INFO [RaftManager id=2] Vote request VoteRequestData(clusterId='Dw6ghXOSQJmC-2XWABqHFA', top
ics=[TopicData(topicName='__cluster_metadata', partitions=[PartitionData(partitionIndex=0, candidateEpoch=148, candida
teId=1, lastOffsetEpoch=147, lastOffset=498474)])]) with epoch 148 is granted (org.apache.kafka.raft.KafkaRaftClient)
[2023-12-06 17:56:14,463] INFO [RaftManager id=2] Completed transition to FollowerState(fetchTimeoutMs=2000, epoch=148
, leaderId=1, voters=[1, 2, 3], highWatermark=Optional[LogOffsetMetadata(offset=498473, metadata=Optional.empty)], fet
chingSnapshot=Optional.empty) from Voted(epoch=148, votedId=1, voters=[1, 2, 3], electionTimeoutMs=1334) (org.apache.k
afka.raft.QuorumState)
[2023-12-06 17:56:14,910] INFO [RaftManager id=2] Node 3 disconnected. (org.apache.kafka.clients.NetworkClient)
[2023-12-06 17:56:14,910] INFO [BrokerToControll                                   ame=heartbeat] Node 3 disconnected. (org.apache
.kafka.clients.NetworkClient)
[2023-12-06 17:56:14,910] INFO [BrokerToControll                                   ame=forwarding] Node 3 disconnected. (org.apach
e.kafka.clients.NetworkClient)
[2023-12-06 17:56:15,680] INFO [BrokerToControllerChannelManager id=2 name=heartbeat] Node 3 disconnected. (org.apache
.kafka.clients.NetworkClient)
[2023-12-06 17:56:15,689] WARN [BrokerToControllerChannelManager id=2 name=heartbeat] Connection to node 3 (localhost/
127.0.0.1:9097) could not be established. Broker may not be available. (org.apache.kafka.clients.NetworkClient)
[2023-12-06 17:56:15,690] INFO [broker-2-to-controller-heartbeat-channel-manager]: Recorded new controller, from now o
n will use node localhost:9093 (id: 1 rack: null) (kafka.server.BrokerToControllerRequestThread)
^C[2023-12-06 17:56:23,373] INFO Terminating process due to signal SIGINT (org.apache.kafka.common.utils.LoggingSignal
Handler)
[2023-12-06 17:56:23,375] INFO [BrokerServer id=2] Transition from STARTED to SHUTTING_DOWN (kafka.server.BrokerServer
)
[2023-12-06 17:56:23,377] INFO [BrokerServer id=2] shutting down (kafka.server.BrokerServer)
[2023-12-06 17:56:23,378] INFO [BrokerLifecycleManager id=2] Beginning controlled shutdown. (kafka.server.BrokerLifecy
cleManager)
```

Control + C

1.25x 1:50 / 3:07

62. Trying how Kafka Producer Delivery & Request Timeout works

```
application.properties
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeria
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000
```

```
Console
with correlation id 74 on topic-partition topic2-0, retrying (2147483585 attempts left). Error: N
with correlation id 75 on topic-partition topic2-0, retrying (2147483584 attempts left). Error: N
with correlation id 76 on topic-partition topic2-0, retrying (2147483583 attempts left). Error: N
with correlation id 77 on topic-partition topic2-0, retrying (2147483582 attempts left). Error: N
with correlation id 78 on topic-partition topic2-0, retrying (2147483581 attempts left). Error: N
with correlation id 79 on topic-partition topic2-0, retrying (2147483580 attempts left). Error: N
with correlation id 80 on topic-partition topic2-0, retrying (2147483579 attempts left). Error: N
with correlation id 81 on topic-partition topic2-0, retrying (2147483578 attempts left). Error: N
with correlation id 82 on topic-partition topic2-0, retrying (2147483577 attempts left). Error: N
with correlation id 83 on topic-partition topic2-0, retrying (2147483576 attempts left). Error: N
with correlation id 84 on topic-partition topic2-0, retrying (2147483575 attempts left). Error: N
with correlation id 85 on topic-partition topic2-0, retrying (2147483574 attempts left). Error: N
with correlation id 86 on topic-partition topic2-0, retrying (2147483573 attempts left). Error: N
with correlation id 87 on topic-partition topic2-0, retrying (2147483572 attempts left). Error: N
with correlation id 88 on topic-partition topic2-0, retrying (2147483571 attempts left). Error: N
with correlation id 89 on topic-partition topic2-0, retrying (2147483570 attempts left). Error: N
with correlation id 90 on topic-partition topic2-0, retrying (2147483569 attempts left). Error: N
with correlation id 91 on topic-partition topic2-0, retrying (2147483568 attempts left). Error: N
with correlation id 92 on topic-partition topic2-0, retrying (2147483567 attempts left). Error: N
with correlation id 93 on topic-partition topic2-0, retrying (2147483566 attempts left). Error: N
with correlation id 94 on topic-partition topic2-0, retrying (2147483565 attempts left). Error: N
with correlation id 95 on topic-partition topic2-0, retrying (2147483564 attempts left). Error: N
with correlation id 96 on topic-partition topic2-0, retrying (2147483563 attempts left). Error: N
with correlation id 97 on topic-partition topic2-0, retrying (2147483562 attempts left). Error: N
```

1.25x 2:25 / 3:07

62. Trying how Kafka Producer Delivery & Request Timeout works

```
application.properties
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeria
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000
```

```
Console
pic-partition topic2-0, retrying (2147483518 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483517 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483516 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483515 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483514 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483513 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483512 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483511 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483510 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483509 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483508 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483507 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483506 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483505 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483504 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483503 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483502 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483501 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483500 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483499 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483498 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483497 attempts left). Error: NOT_ENOUGH_REPLICAS
pic-partition topic2-0, retrying (2147483496 attempts left). Error: NOT_ENOUGH_REPLICAS
```

Is trying to send Kafka message again and again, and the maximum time that I have configured for this

1.25x 2:27 / 3:07

62. Trying how Kafka Producer Delivery & Request Timeout works

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSerie
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
```

```
        at org.apache.tomcat.util.net.NioEndpoint$SocketProcessor.doRun(NioEndpoint.java:1744) ~[t
        at org.apache.tomcat.util.net.SocketProcessorBase.run(SocketProcessorBase.java:52) ~[tomca
        at org.apache.tomcat.util.threads.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:119
        at org.apache.tomcat.util.threads.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:65
        at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61) ~[to
        at java.base/java.lang.Thread.run(Thread.java:833) ~[na:na]
Caused by: org.springframework.kafka.core.KafkaProducerException: Failed to send
        at org.springframework.kafka.core.KafkaTemplate.lambda$buildCallback$9(KafkaTemplate.java:
        at org.springframework.kafka.core.DefaultKafkaProducerFactory$CloseSafeProducer$1.onComple
        at org.apache.kafka.clients.producer.KafkaProducer$AppendCallbacks.onCompletion(KafkaProdu
        at org.apache.kafka.clients.producer.internals.ProducerBatch.completeFutureAndFireCallback
        at org.apache.kafka.clients.producer.internals.ProducerBatch.done(ProducerBatch.java:234)
        at org.apache.kafka.clients.producer.internals.ProducerBatch.completeExceptionally(Produce
        at org.apache.kafka.clients.producer.internals.Sender.failBatch(Sender.java:798) ~[kafka-c
        at org.apache.kafka.clients.producer.internals.Sender.failBatch(Sender.java:787) ~[kafka-c
        at org.apache.kafka.clients.producer.internals.Sender.sendProducerData(Sender.java:422) ~[
        at org.apache.kafka.clients.producer.internals.Sender.runOnce(Sender.java:343) ~[kafka-cli
        at org.apache.kafka.clients.producer.internals.Sender.run(Sender.java:247) ~[kafka-clients
        ... 1 common frames omitted
Caused by: org.apache.kafka.common.errors.TimeoutException: Expiring 1 record(s) for topic2-0:1200

2023-12-06T17:59:04.626-05:00  INFO 23890 --- [ad | producer-1] o.a.k.c.p.internals.TransactionMan
2023-12-06T17:59:04.626-05:00  INFO 23890 --- [ad | producer-1] o.a.k.c.p.internals.TransactionMan
```

And if I scroll to the left, I see Java Exception.

1.25x 2:52 / 3:07

```
server.port=0

spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerialize
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSe
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000
```

is that to configure key and value serializers, I need to type complete name of the class, including

63. Kafka Producer Spring Bean Configuration

```java
@Configuration
public class KafkaConfig {

    @Value("${spring.kafka.producer.bootstrap-servers}")
    private String bootstrapServers;

    @Value("${spring.kafka.producer.key-serializer}")
    private String keySerializer;

    @Value("${spring.kafka.producer.value-serializer}")
    private String valueSerializer;

    @Value("${spring.kafka.producer.acks}")
    private String acks;

    @Value("${spring.kafka.producer.properties.delivery.timeout.ms}")
    private String deliveryTimeout;

    @Value("${spring.kafka.producer.properties.linger.ms}")
    private String linger;

    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
    private String requestTimeout;

    Map<String, Object> producerConfigs() {
        Map<String, Object> config = new HashMap<>();

        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        config.put(ProducerConfig.ACKS_CONFIG, acks);
        config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

        return config;
    }
```

All right, so I've added other Kafka producer related configuration properties that I have in my application

1.25x 3:53 / 8:24

63. Kafka Producer Spring Bean Configuration

```java
    @Value("${spring.kafka.producer.properties.linger.ms}")
    private String linger;

    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
    private String requestTimeout;

    Map<String, Object> producerConfigs() {
        Map<String, Object> config = new HashMap<>();

        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        config.put(ProducerConfig.ACKS_CONFIG, acks);
        config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

        return config;
    }

    @Bean
    ProducerFactory<String, ProductCreatedEvent> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
    NewTopic createTopic() {
        return TopicBuilder.name("product-created-events-topic")
                .partitions(3)
                .replicas(3)
                .configs(Map.of("min.insync.replicas","2"))
                .build();
    }

}
```

Kafka producer instances in Spring Framework.

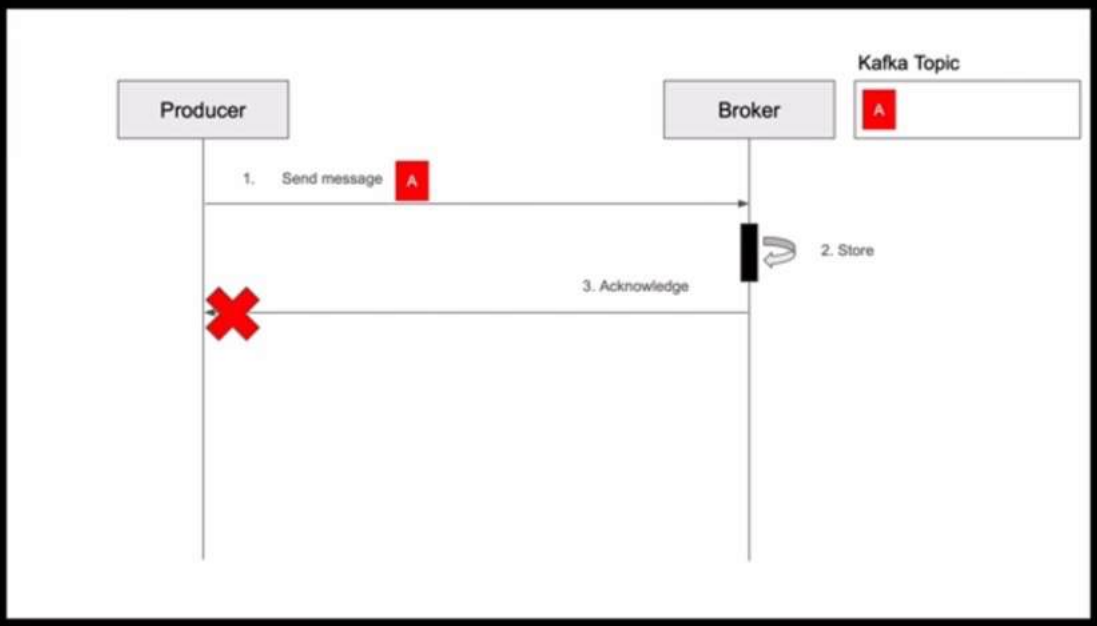
Pause

1.25x 5:02 / 8:24

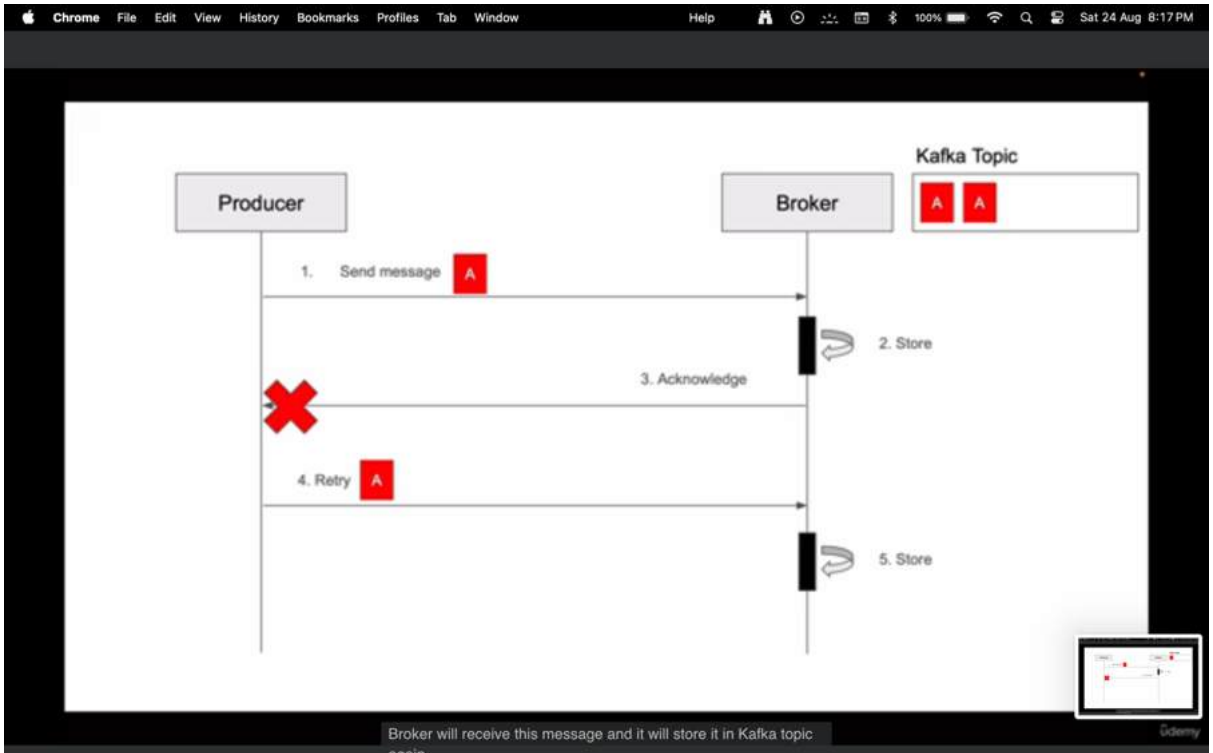
63. Kafka Producer Spring Bean Configuration

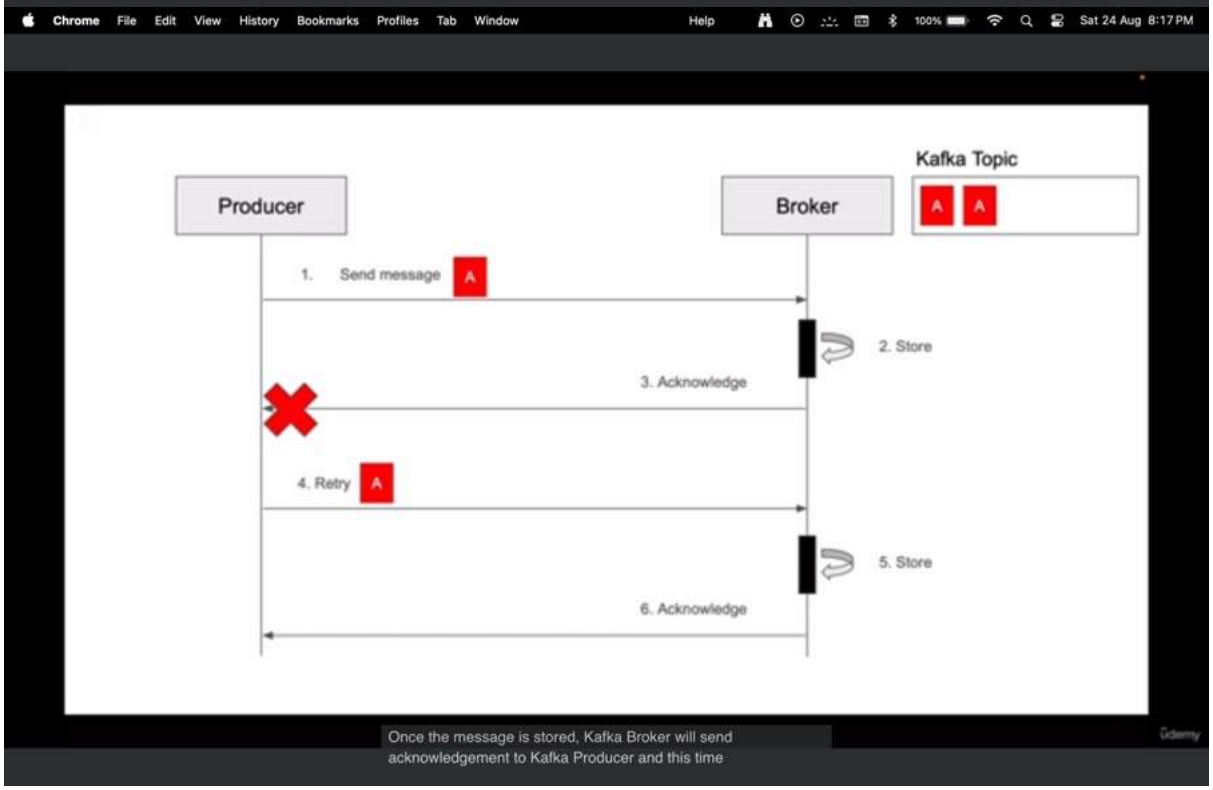
```
private String linger;

@Value("${spring.kafka.producer.properties.request.timeout.ms}")
private String requestTimeout;

Map<String, Object> producerConfigs() {
    Map<String, Object> config = new HashMap<>();

    config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
    config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
    config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
    config.put(ProducerConfig.ACKS_CONFIG, acks);
    config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
    config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
    config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

    return config;
}

@Bean
ProducerFactory<String, ProductCreatedEvent> producerFactory() {
    return new DefaultKafkaProducerFactory<>(producerConfigs());
}

@Bean
KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate() {
    return new KafkaTemplate<String, ProductCreatedEvent>(producerFactory());
}

@Bean
NewTopic createTopic() {
    return TopicBuilder.name("product-created-events-topic")
            .partitions(3)
            .replicas(3)
```

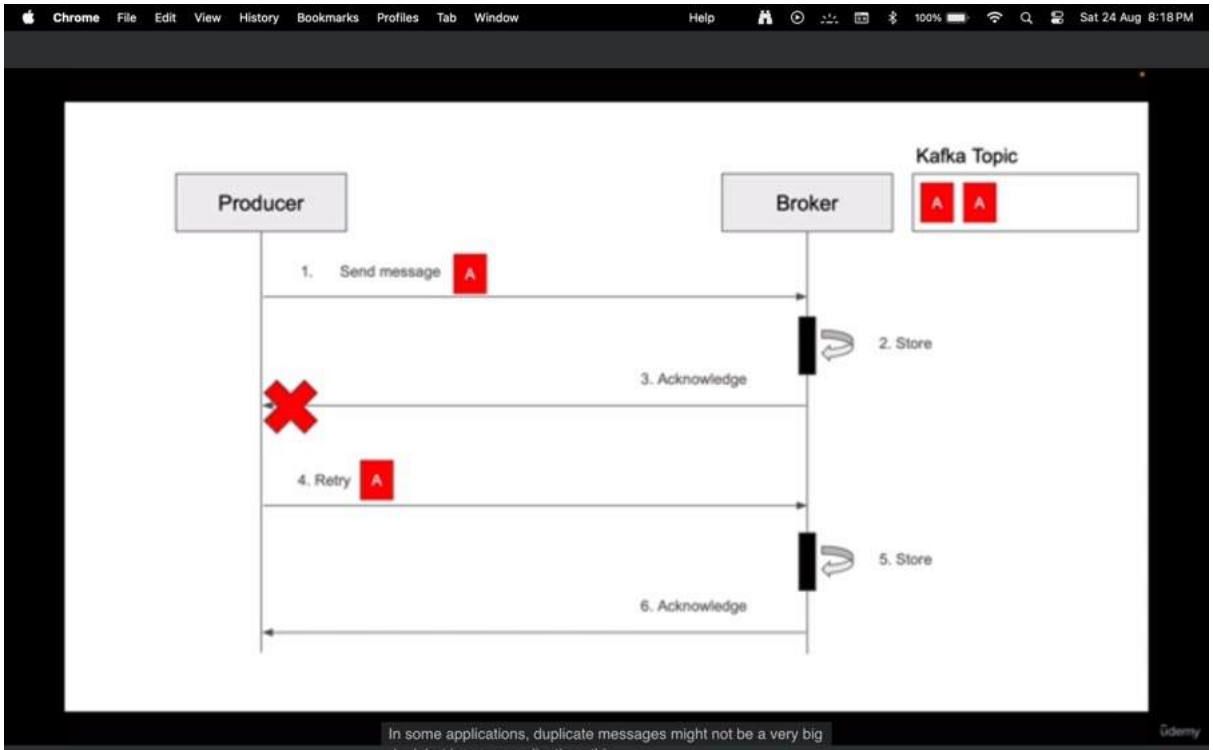
This way we can then inject Kafka template bin into other classes that need to send messages to Kafka



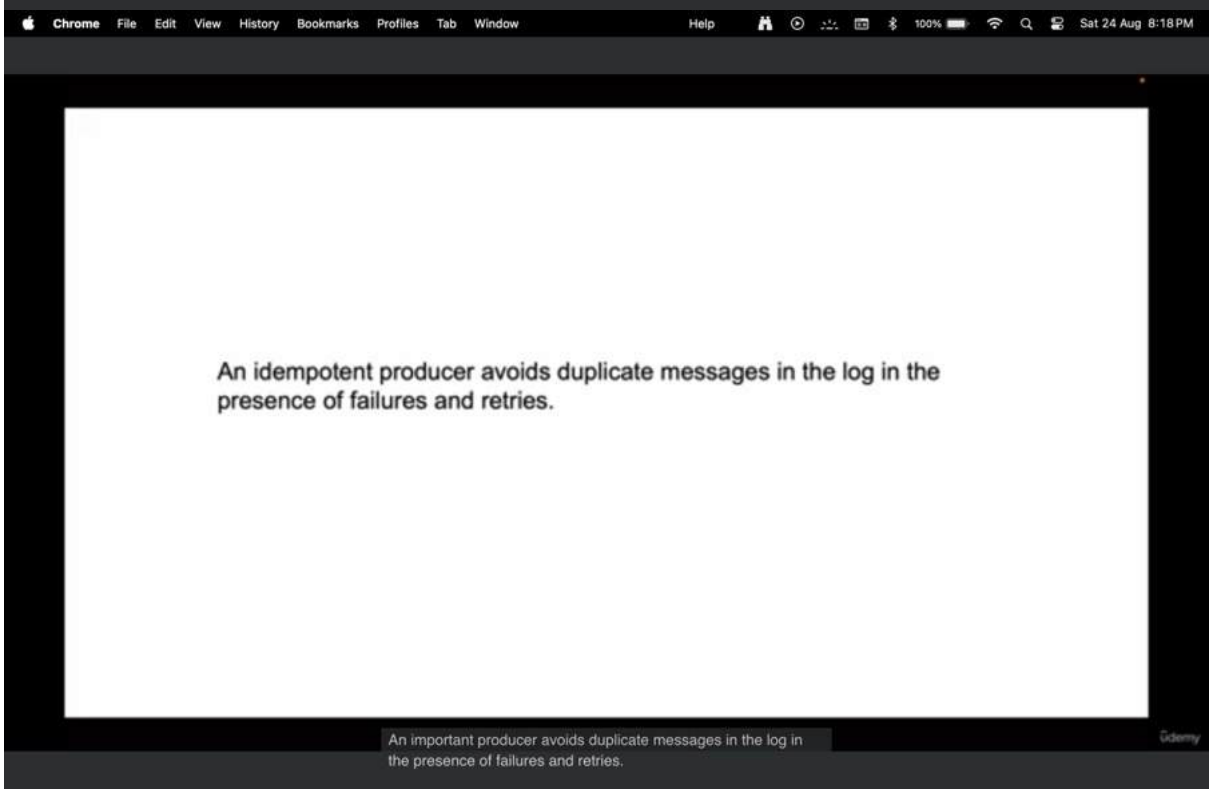
If producer retries are configured, then producer will send the same message again.


Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:17 PM
Kafka Topic
Producer
Broker
A A
1. Send message A
2. Store
3. Acknowledge
4. Retry A
5. Store
Broker will receive this message and it will store it in Kafka topic again.

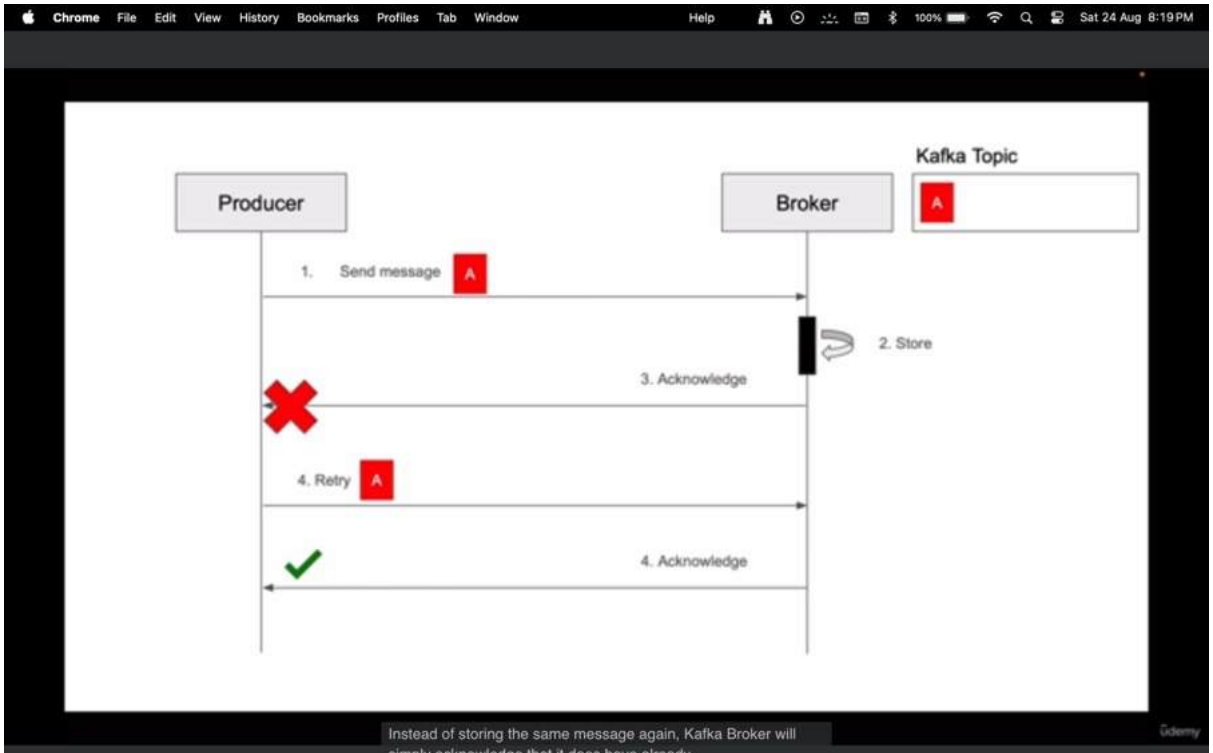

Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:17 PM
Kafka Topic
Producer
Broker
A A
1. Send message A
2. Store
3. Acknowledge
4. Retry A
5. Store
6. Acknowledge
Once the message is stored, Kafka Broker will send acknowledgement to Kafka Producer and this time



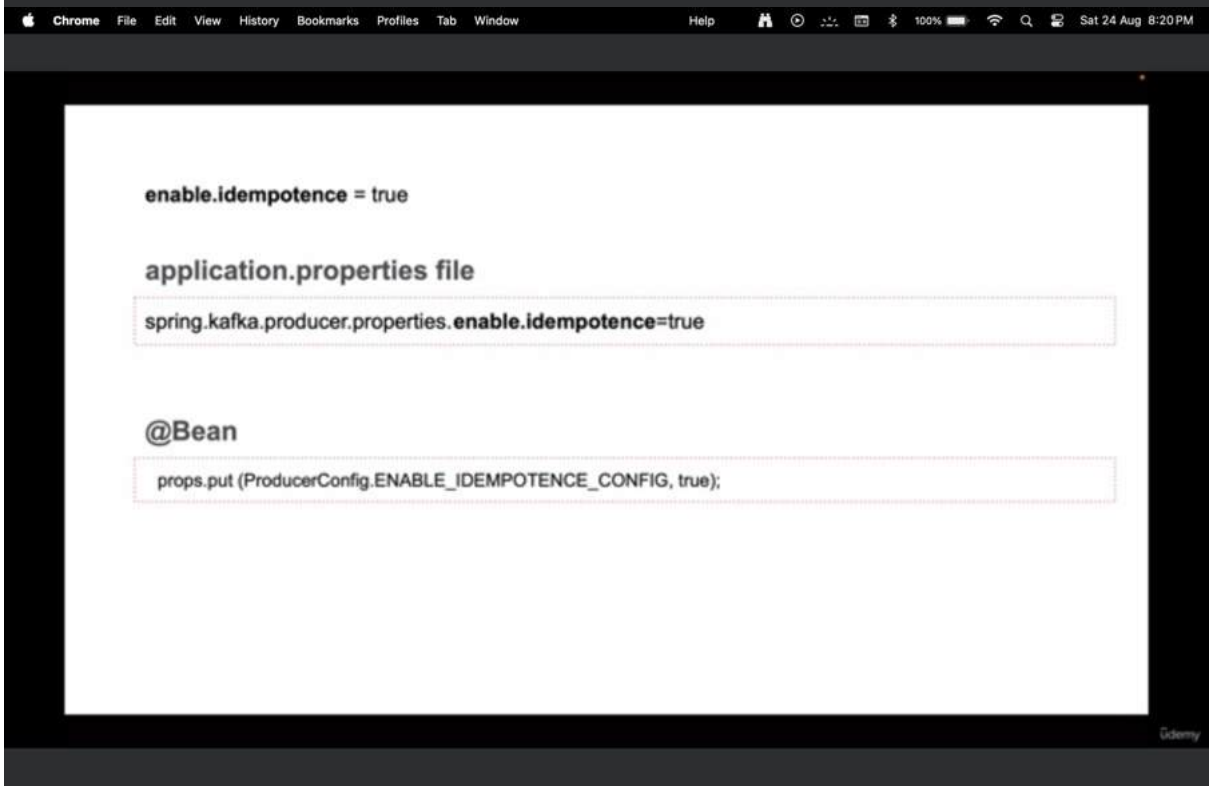
In some applications, duplicate messages might not be a very big deal, but in some applications this



An important producer avoids duplicate messages in the log in the presence of failures and retries.



Instead of storing the same message again, Kafka Broker will simply acknowledge that it does have already



Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:20 PM

acks=all

retries=2147483647

max.in.flight.requests.per.connection=5

If you are using Application.properties file, then for Idempotence to work, acknowledgment property

Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:21 PM

acks=all

retries=2147483647

max.in.flight.requests.per.connection=5

application.properties file

spring.kafka.producer.properties.**acks**=all

spring.kafka.producer.properties.**retries**=2147483647

spring.kafka.producer.properties.**max.in.flight.requests.per.connection**=5

producer can send up to five requests or batches of messages to broker at the same time without waiting

Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:21 PM

acks=all

retries=2147483647

max.in.flight.requests.per.connection=5

application.properties file

```
spring.kafka.producer.properties.acks=all
spring.kafka.producer.properties.retries=2147483647
spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true
```

Udemy

Idempotence configuration property.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sat 24 Aug 8:22 PM

acks=all

retries=2147483647

max.in.flight.requests.per.connection=5

application.properties file

```
spring.kafka.producer.properties.acks=all
spring.kafka.producer.properties.retries=2147483647
spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true
```

ConfigException

And in the following lessons we will update Java code of our Kafka producer to enable Idempotence.

Udemy

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerialize
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSe
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.enable.idempotence=true
spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
```

if I run this application and try to produce a message, it should work.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/KafkaConfig.java - Spring Tool Suite 4

```
    private String acks;

    @Value("${spring.kafka.producer.properties.delivery.timeout.ms}")
    private String deliveryTimeout;

    @Value("${spring.kafka.producer.properties.linger.ms}")
    private String linger;

    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
    private String requestTimeout;

    @Value("${spring.kafka.producer.properties.enable.idempotence}")
    private boolean idempotence;

    @Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
    private Integer inflightRequests;

    Map<String, Object> producerConfigs() {
        Map<String, Object> config = new HashMap<>();

        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        config.put(ProducerConfig.ACKS_CONFIG, acks);
        config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
        config.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);

        return config;
    }

    @Bean
    ProducerFactory<String, ProductCreatedEvent> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
```

And next I will add this configuration property to a map object that I have here.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/KafkaConfig.java - Spring Tool Suite 4

```java
    private String acks;

    @Value("${spring.kafka.producer.properties.delivery.timeout.ms}")
    private String deliveryTimeout;

    @Value("${spring.kafka.producer.properties.linger.ms}")
    private String linger;

    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
    private String requestTimeout;

    @Value("${spring.kafka.producer.properties.enable.idempotence}")
    private boolean idempotence;

    @Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
    private Integer inflightRequests;

    Map<String, Object> producerConfigs() {
        Map<String, Object> config = new HashMap<>();

        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        config.put(ProducerConfig.ACKS_CONFIG, acks);
        config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
        config.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        config.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

        return config;
    }

    @Bean
    ProducerFactory<String, ProductCreatedEvent> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }
```

per connection must not be greater than five and five is what I have in my application properties file,

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/KafkaConfig.java - Spring Tool Suite 4

```java
    private String acks;

    @Value("${spring.kafka.producer.properties.delivery.timeout.ms}")
    private String deliveryTimeout;

    @Value("${spring.kafka.producer.properties.linger.ms}")
    private String linger;

    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
    private String requestTimeout;

    @Value("${spring.kafka.producer.properties.enable.idempotence}")
    private boolean idempotence;

    @Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
    private Integer inflightRequests;

    Map<String, Object> producerConfigs() {
        Map<String, Object> config = new HashMap<>();

        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        config.put(ProducerConfig.ACKS_CONFIG, acks);
        config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
        config.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        config.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);
        config.put(ProducerConfig.RETRIES_CONFIG, Integer.MAX_VALUE);

        return config;
    }

    @Bean
    ProducerFactory<String, ProductCreatedEvent> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }
```

I'm not going to change this value and I will leave it as is.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/KafkaConfig.java - Spring Tool Suite 4

```java
private String acks;

@Value("${spring.kafka.producer.properties.delivery.timeout.ms}")
private String deliveryTimeout;

@Value("${spring.kafka.producer.properties.linger.ms}")
private String linger;

@Value("${spring.kafka.producer.properties.request.timeout.ms}")
private String requestTimeout;

@Value("${spring.kafka.producer.properties.enable.idempotence}")
private boolean idempotence;

@Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
private Integer inflightRequests;

Map<String, Object> producerConfigs() {
    Map<String, Object> config = new HashMap<>();

    config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
    config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
    config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
    config.put(ProducerConfig.ACKS_CONFIG, acks);
    config.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
    config.put(ProducerConfig.LINGER_MS_CONFIG, linger);
    config.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
    config.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
    config.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);
    //config.put(ProducerConfig.RETRIES_CONFIG, Integer.MAX_VALUE);

    return config;
}

@Bean
ProducerFactory<String, ProductCreatedEvent> producerFactory() {
    return new DefaultKafkaProducerFactory<>(producerConfigs());
}
```

This should enable Idempotence for my Kafka producer.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

```properties
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeria
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.enable.idempotence=true
spring.kafka.producer.properties.max.in.flight.requests.per.connection=6
```

```
cer.KafkaProducer      : [Producer clientId=producer-1] Instantiated an idempotent producer.
ils.AppInfoParser      : Kafka version: 3.6.1
ils.AppInfoParser      : Kafka commitId: 5e3c2b738d253ff5
ils.AppInfoParser      : Kafka startTimeMs: 1706484830311
ients.Metadata         : [Producer clientId=producer-1] Cluster ID: TAT5ntAtTXWf64YmxcJI2A
.TransactionManager    : [Producer clientId=producer-1] ProducerId set to 1003 with epoch 0
ductServiceImpl        : Partition: 1
ductServiceImpl        : Topic: product-created-events-topic
ductServiceImpl        : Offset: 2
ductServiceImpl        : ***** Returning product id
```

6 or 10 for example.

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

*application.properties

```
server.port=0

spring.kafka.producer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeria
spring.kafka.producer.acks=all
#spring.kafka.producer.retries=10
#spring.kafka.producer.properties.retry.backoff.ms=1000

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.enable.idempotence=true
spring.kafka.producer.properties.max.in.flight.requests.per.connection=6
```

```
cer.KafkaProducer      : [Producer clientId=producer-1] Instantiated an idempotent producer.
ils.AppInfoParser      : Kafka version: 3.6.1
ils.AppInfoParser      : Kafka commitId: 5e3c2b738d253ff5
ils.AppInfoParser      : Kafka startTimeMs: 1706484830311
ients.Metadata         : [Producer clientId=producer-1] Cluster ID: TAT5ntAtTXWf64YmxcJI2A
.TransactionManager    : [Producer clientId=producer-1] ProducerId set to 1003 with epoch 0
ductServiceImpl        : Partition: 1
ductServiceImpl        : Topic: product-created-events-topic
ductServiceImpl        : Offset: 2
ductServiceImpl        : ***** Returning product id
```

So this should cause configuration exception, because I explicitly told Kafka that I want to enable

Postman

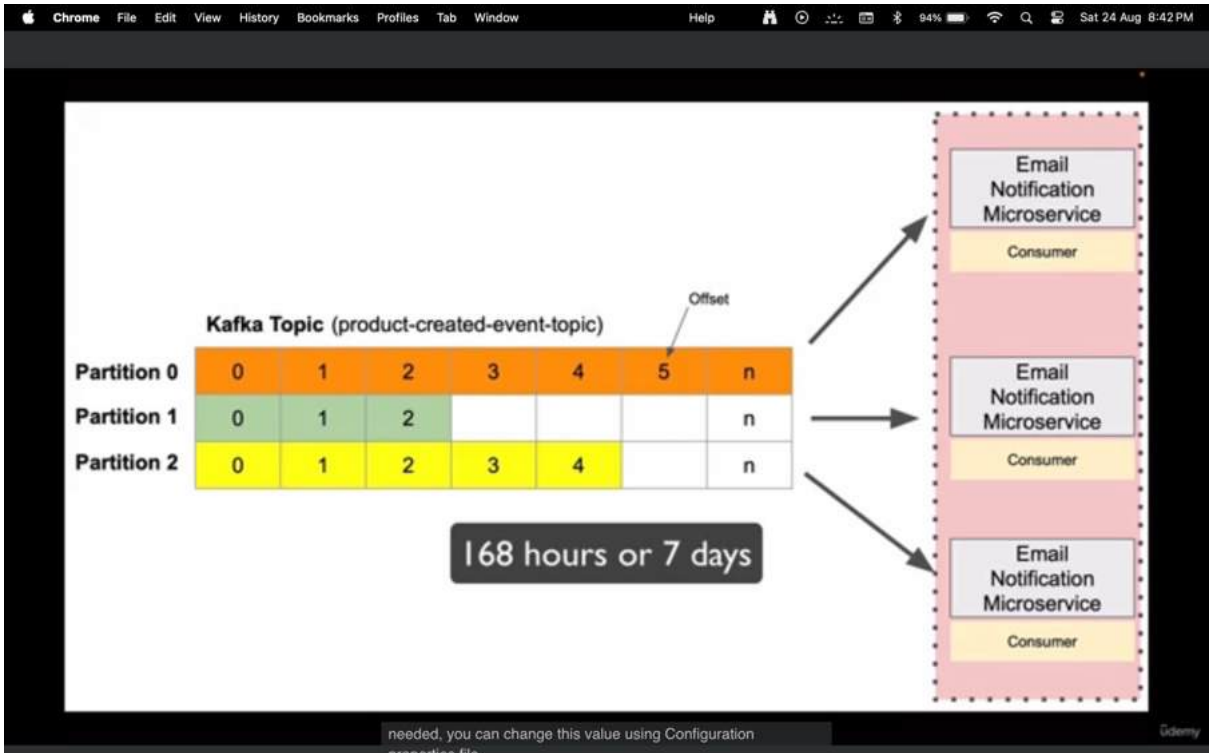
POST http://localhost:50283/products

```
{
    "title":"iPad Pro",
    "price":1250,
    "quantity":19
}
```

Status: 500 Internal Server Error Time: 202 ms Size: 325 B

```
{
    "timestamp": "2024-01-28T23:35:24.953+00:00",
    "message": "Must set max.in.flight.requests.per.connection to at most 5 to use the idempotent producer.",
    "details": "/products"
```

The error message says that we must set the maximum in-flight requests per connection to at most five,



needed, you can change this value using Configuration properties file.

70. Kafka Consumer Configuration Properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
spring.kafka.consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=*
```

But in one of the following lessons, I will come back here and I will replace Asterix with a specific

Pause 1.5x 6:33 / 6:44

71. Kafka Consumer with @KafkaEventListener and @KafkaHandler annotations

@KafkaListener can be configured to listen to multiple topics.
For example,
@KafkaListener(topics = {"topic1", "topic2"})
A single Kafka Consumer can consume messages from multiple topics.

```
@Component
public class ProductCreatedEventHandler {

    @KafkaListener(topics="product-created-events-topic")
    public void handle() {

    }

}
```

Now to make this method handle product created event I'll need to make it accept product created event

1.5x 2:02 / 4:44

Confirmation of Redundancy JC - Inbox - R725693@EMEA.AD.JPMORGANCHASE.com

Delete Archive Move Flag Mark as Unread Sync Report ...

Confirmation of Redundancy JC

HR EDSE UK Solutions <hr.edse.uk.solutions@chase.com>
To: Patel, Parul X2
Tuesday 20 August 2024 at 4:07 PM

Alumni Network One... 57.6 KB | EMEA Talent Redepl... 203.1 KB | Accessing Stock Aw... 158.4 KB | UK Professional Pro... 288.1 KB

Download All · Preview All

Retention: JPMC_Inbox_180 Expires: 16/02/2025.

Hi Parul,

Following your end of consultation meeting held today where it was explained that your current role was confirmed redundant, this email represents notice of termination. Your employment will terminate by reason of redundancy as from 19 September 2024 unless a suitable alternative opportunity is identified.

I wanted to reach out to see if you had any initial questions and ensure you have my contact details.
As previously discussed, the redeployment team are still available to assist you in looking for an alternative role. Should you wish to attend a workshop, please email 'Redeployment Workshops' or (externally) redeployment.workshops@jpmorgan.com to register your interest.

Additionally, outplacement support is available which can be used to help you with your external job search. Please do take a look at the handout provided for more information on how to utilise this service.
I would also like to remind you of our Employee Assistance Program should you wish to speak to a trained adviser on any personal matters. They can be reached 24 hours a day on **0800 141 2839.**

Returning your Settlement Agreement:
When returning your settlement agreement, please ensure you have had it reviewed and the Adviser's Certificate signed by an independent solicitor. Once you have received independent legal advice and are happy with the terms of the settlement agreement, please can you return the signed settlement agreement by email to this email address marking it for my attention.
It is likely that the solicitor will require a copy of your contract in advance of your appointment with them. If you do not have a copy saved, we would suggest that you request this by going to the new me@jpmc and raising a case via me@jpmc
In terms of signing the document once you have taken independent legal advice, please print out the document, sign it and scan it back to the Company by email. Please instruct your Solicitor to do the same with their certificate. Please direct all correspondence to this address, marking it for my attention.
If you do not have any printing and scanning capabilities at home, the Company can accept a copy of the agreement which has been signed on your behalf by your legal adviser. To do this, please can you send a new** email to your legal adviser stating the following "*I have read, understood and accept the terms of the settlement agreement. Please sign this agreement on my behalf. Please note that I shall be sending a copy of this authorisation email to JP Morgan as evidence that I have authorised you, NAME OF LAWYER of NAME OF LAW FIRM to sign the settlement agreement on my behalf*".
Please complete the text in red as appropriate.
** Please note that it is very important that you send a new email rather than replying to an earlier chain which may contain confidential and legally privileged information between you and your legal

71. Kafka Consumer with @KafkaEventListener and @KafkaHandler annotations

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.kafka.annotation.KafkaHandler;
import org.springframework.kafka.annotation.KafkaListener;
import org.springframework.stereotype.Component;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreateEvent.getTitle());
    }

}
```

And then the following lessons.

73. Adding "core" project as a dependency

```pom.properties
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSch
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/mave
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.appsdeveloperblog.ws</groupId>
    <artifactId>core</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>core</name>
    <description>A shared CORE library</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
    </dependencies>


</project>
```

These are Maven coordinates of this project, and I will use them to add this project as a dependency

73. Adding "core" project as a dependency

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/mave
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.2.0</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.appsdeveloperblog.ws</groupId>
    <artifactId>ProductsMicroservice</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>ProductsMicroservice</name>
    <description>Products Microservice</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>

        <dependency>
            <groupId>com.appsdeveloperblog.ws</groupId>
            <artifactId>core</artifactId>
            <version>0.0.1-SNAPSHOT</version>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.kafka</groupId>
            <artifactId>spring-kafka</artifactId>
        </dependency>
```

Now that I have added core dependency to products microservice, I will delete from this project the

1.25x 1:08 / 4:46

73. Adding "core" project as a dependency

```
package com.appsdeveloperblog.ws.products.service;

import java.util.UUID;

@Service
public class ProductServiceImpl implements ProductService {

    KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate;
    private final Logger L
    public ProductServiceI                                  late) {
        this.kafkaTemplate
    }

    @Override
    public String createPr                                  ception {

        String productId =

        // TODO: Persist P                                  Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send("product-created-events-topic",productId, productCreatedEv

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());
        LOGGER.info("Offset: " + result.getRecordMetadata().offset());

        LOGGER.info("***** Returning product id");
```

ProductCreatedEvent cannot be resolved to a type

8 quick fixes available:

- Import 'ProductCreatedEvent' (com.appsdeveloperblog.ws.core)
- Add com.appsdeveloperblog.ws:core:0.0.1-SNAPSHOT as dependency
- Create class 'ProductCreatedEvent'
- Create record 'ProductCreatedEvent'
- Create interface 'ProductCreatedEvent'
- Create enum 'ProductCreatedEvent'
- Add type parameter 'ProductCreatedEvent' to 'ProductServiceImpl'

Most likely a little glitch in my development environment, so I will use this option to import product

1.25x 2:53 / 4:46

73. Adding "core" project as a dependency

```
<modelVersion>4.0.0</modelVersion>
<parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.2.0</version>
    <relativePath /> <!-- lookup parent from repository -->
</parent>
<groupId>com.appsdeveloperblog.ws</groupId>
<artifactId>EmailNotificationMicroservice</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>EmailNotificationMicroservice</name>
<description>Email Notification Microservice</description>
<properties>
    <java.version>17</java.version>
</properties>
<dependencies>
    <dependency>
        <groupId>com.appsdeveloperblog.ws</groupId>
        <artifactId>core</artifactId>
        <version>0.0.1-SNAPSHOT</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.kafka</groupId>
        <artifactId>spring-kafka</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
```

Now I will save changes to Pom.xml file of email notification microservice and will update this project

1.25x 3:54 / 4:46

73. Adding "core" project as a dependency

```
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("R
    }

}
```

ProductCreatedEvent cannot be resolved to a type

10 quick fixes available:

- Import 'ProductCreatedEvent' (com.appsdeveloperblog.ws.core)
- Add com.appsdeveloperblog.ws:core:0.0.1-SNAPSHOT as dependency
- Create class 'ProductCreatedEvent'
- Create record 'ProductCreatedEvent'
- Create interface 'ProductCreatedEvent'
- Change to 'ProductCreatedEventHandler' (com.appsdeveloperblog.ws.emailnotification.handler)
- Create enum 'ProductCreatedEvent'

So I will choose import product created event.

1.25x 4:17 / 4:46

73. Adding "core" project as a dependency

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

No errors in email notification microservice class as well.

74. @KafkaHandler: Trying how it works

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

So let's try now.

74. @KafkaHandler: Trying how it works

```
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

```
2023-12-10T10:26:42.385-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:42.385-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.410-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.418-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.455-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.455-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.458-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.470-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.470-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.470-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.473-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.473-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.473-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordin
2023-12-10T10:26:45.489-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionSta
2023-12-10T10:26:45.490-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionSta
2023-12-10T10:26:45.490-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionSta
2023-12-10T10:26:45.515-05:00  INFO 5393 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContain
```

1.25x 1:02 / 2:30

74. @KafkaHandler: Trying how it works

```
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

```
on.utils.AppInfoParser     : Kafka version: 3.6.0
on.utils.AppInfoParser     : Kafka commitId: 60e845626d8a465a
on.utils.AppInfoParser     : Kafka startTimeMs: 1702221980500
on.utils.AppInfoParser     : App info kafka.admin.client for adminclient-1 unregistered
.common.metrics.Metrics    : Metrics scheduler closed
.common.metrics.Metrics    : Closing reporter org.apache.kafka.common.metrics.JmxReporter
.common.metrics.Metrics    : Metrics reporters closed
ed.tomcat.TomcatWebServer  : Tomcat started on port [redacted] (http) with context path ''
tsMicroserviceApplication  : Started ProductsMicroserviceApplication in 1.725 seconds (process run
```

Next I will open postman and we'll send the Http request.

1.25x 1:10 / 2:30

74. @KafkaHandler: Trying how it works

POST http://localhost:50627/products

Send

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 450 ms Size: 205 B

```
88d9d024-9c4f-40cc-8dec-7de5ccc8cd93
```

And in response I received product ID.

1:32 / 2:30

```
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

```
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] (Re
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Suc
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fin
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Suc
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Not
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Add
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
nator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Fou
ate    : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Res
ate    : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Res
ate    : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Res
ner    : product-created-events: partitions assigned: [product-created-events-topic-0, product-cre
er     : Received a new event: iPhone 11
```

75. Kafka Consumer Spring Bean Configuration

```
application.properties
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
spring.kafka.consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=*
```

It is not very convenient to type the entire package name, and it is very easy to make a typo.

```
*application.properties
nsumer.bootstrap-servers=localhost:9092,localhost:9094
nsumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
nsumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
nsumer.group-id=product-created-events
nsumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.ws.core
```

This way it is going to be more secure, because only classes that belong to this package can be converted

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

application.properties | KafkaConsumerConfiguration.java

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import com.fasterxml.jackson.databind.deser.std.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;


    @Bean
    public ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES, environment.getProperty("spring.kafka.consumer.properties.spring.json.trustec
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

}
```

Now that we have Kafka consumer configuration properties created here in Spring Bin, we can comment

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

application.properties | *KafkaConsumerConfiguration.java

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;

import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;


    @Bean
    public ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES, environment.getProperty("spring.kafka.consumer.properties.spring.json.trust
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

}
```

org.apache.kafka.common.serialization.StringDeserializer

Org Apache Kafka command serialization package.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

application.properties | *KafkaConsumerConfiguration.java

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory) {

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        return factory;
    }
}
```

So with this method we created another special object that will be used to make Kafka listener container

Postman — POST http://localhost:52775/products

Body (raw, JSON):

```
{
    "title":"iPhone 11",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 759 ms Size: 205 B

Response:

```
27f9734c-1b84-4639-ab15-78aba0cd4936
```

And I have a successful response.

77. Trying if Kafka Consumer Bean Configuration works

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

application.properties | KafkaConsumerConfiguration.java

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
```

Problems | Javadoc | Declaration | Console | Progress

```
--- [           main] EmailNotificationMicroserviceApplication : Started EmailNotificationMicroserviceApplication in 1.421 seconds (proce
--- [ntainer#0-0-C-1] org.apache.kafka.clients.Metadata        : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions assigned: [product-created-events-top
--- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received a new event: iPhone 11
```

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

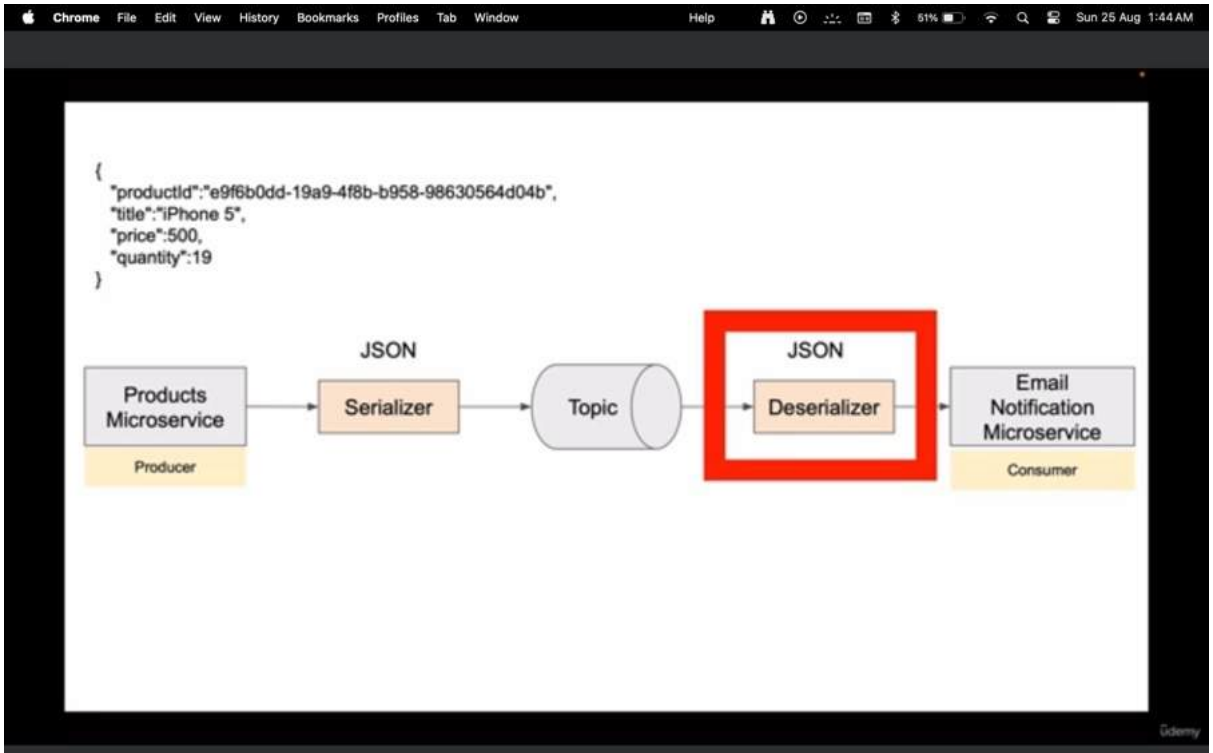
application.properties | KafkaConsumerConfiguration.java

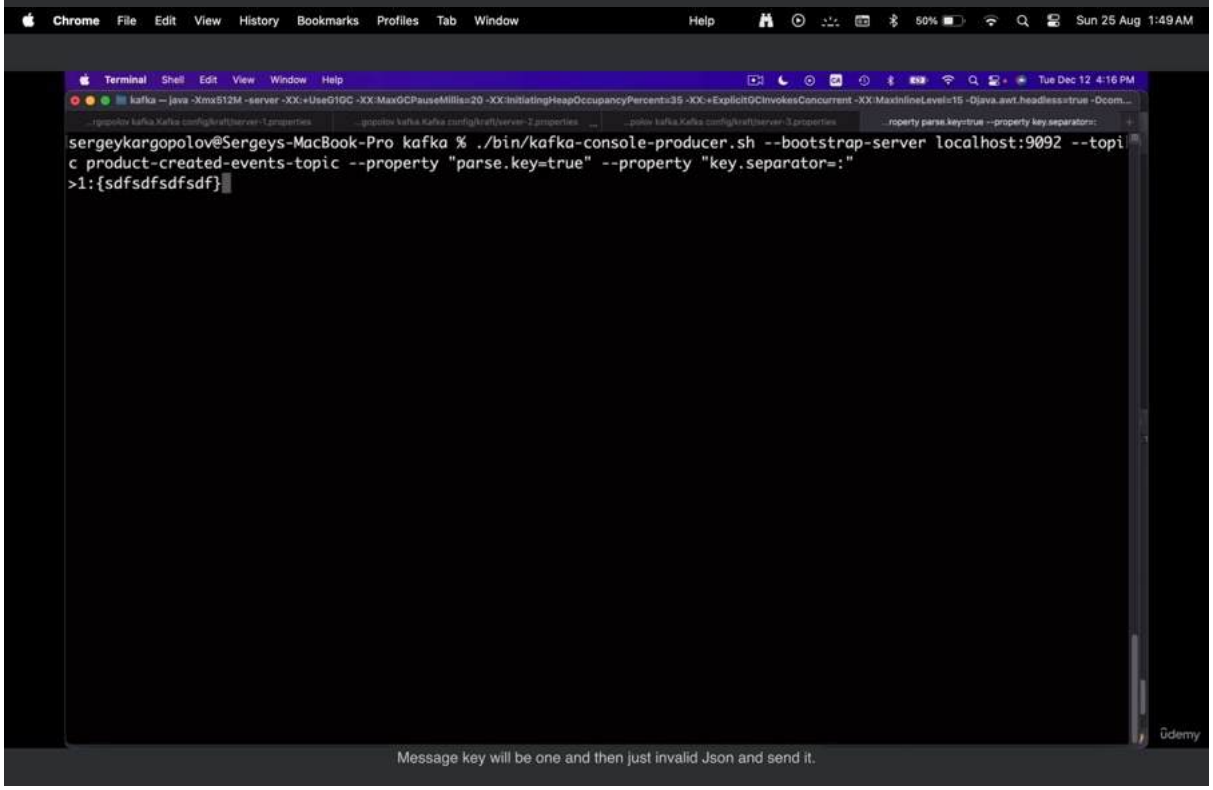
```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
```

Problems | Javadoc | Declaration | Console | Progress

```
--- [ntainer#0-0-C-1] org.apache.kafka.clients.Metadata        : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-events-1, groupId=product-cr
--- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions assigned: [product-created-events-top
--- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received a new event: iPhone 11
```

```
--- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received a new event: iPhone 12
```





Message key will be one and then just invalid Json and send it.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 50% Sun 25 Aug 1:49 AM

SpringToolSuite4 File Edit Navigate Search Project Run Window Help Tue Dec 12 4:16 PM

Event-Driven Microservices with Apache Kafka - Spring Tool Suite 4

Package Explorer
- core
- EmailNotificationMicroservice [boot]
- ProductsMicroservice [boot] [ProductsMicroservice Kafka-

Boot Dashboard
- local
  - EmailNotificationMicroservice [:56550]
  - ProductsMicroservice

Problems Javadoc Declaration Console Progress

```
        at org.apache.kafka.clients.consumer.internals.AbstractFetch.fetchRecords(AbstractFet
        at org.apache.kafka.clients.consumer.internals.AbstractFetch.collectFetch(AbstractFet
        at org.apache.kafka.clients.consumer.KafkaConsumer.pollForFetches(KafkaConsumer.java:
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1186) ~[ka
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1159) ~[ka
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        ... 2 common frames omitted
Caused by: java.lang.IllegalStateException: No type information in headers and no default typ
        at org.springframework.util.Assert.state(Assert.java:76) ~[spring-core-6.1.1.jar:6.1.
        at org.springframework.kafka.support.serializer.JsonDeserializer.deserialize(JsonDese
        at org.apache.kafka.common.serialization.Deserializer.deserialize(Deserializer.java:7
        at org.apache.kafka.clients.consumer.internals.CompletedFetch.parseRecord(CompletedFe
        ... 12 common frames omitted
```

Udemy

And here I see a lot of Java exceptions that are being thrown.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 50% Sun 25 Aug 1:49 AM

SpringToolSuite4 File Edit Navigate Search Project Run Window Help Tue Dec 12 4:16 PM

Event-Driven Microservices with Apache Kafka - Spring Tool Suite 4

Package Explorer
- core
- EmailNotificationMicroservice [boot]
- ProductsMicroservice [boot] [ProductsMicroservice Kafka-

Boot Dashboard
- local
  - EmailNotificationMicroservice [:56550]
  - ProductsMicroservice

Problems Javadoc Declaration Console Progress

```
        at org.apache.kafka.clients.consumer.internals.AbstractFetch.fetchRecords(AbstractFet
        at org.apache.kafka.clients.consumer.internals.AbstractFetch.collectFetch(AbstractFet
        at org.apache.kafka.clients.consumer.KafkaConsumer.pollForFetches(KafkaConsumer.java:
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1186) ~[ka
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1159) ~[ka
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        ... 2 common frames omitted
Caused by: java.lang.IllegalStateException: No type information in headers and no default typ
        at org.springframework.util.Assert.state(Assert.java:76) ~[spring-core-6.1.1.jar:6.1.
        at org.springframework.kafka.support.serializer.JsonDeserializer.deserialize(JsonDese
        at org.apache.kafka.common.serialization.Deserializer.deserialize(Deserializer.java:7
        at org.apache.kafka.clients.consumer.internals.CompletedFetch.parseRecord(CompletedFe
        ... 12 common frames omitted
```

Udemy

And the bad news is that my Kafka consumer is now stuck, and it cannot process any other messages from

81. Kafka Consumer ErrorHandlingDeserializer

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory) {
```

Now this class error handling deserializer.

Pause 1.25x 1:44 / 3:16

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory) {
```

So this new line, it says that when error handling the serializer class is used as a value deserializer,

KafkaConsumerConfiguration.java

```java
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory) {
```

If my Kafka consumer can recover from errors that are caused by deserialization.

POST http://localhost:57953/products

Body

```
{
    "title":"iPhone 12",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 495 ms Size: 205 B

```
04e8ad0d-c6bc-464c-8381-57417b28f865
```

And I get back a successful response.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 41% Sun 25 Aug 2:24 AM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Wed Dec 13 9:52 AM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

Package Explorer
- core
- EmailNotificationMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.emailnotification
      - EmailNotificationMicroserviceApplication.java
      - KafkaConsumerConfiguration.java
    - com.appsdeveloperblog.ws.emailnotification.handler
      - ProductCreatedEventHandler.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
    - target
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
- ProductsMicroservice [boot] [ProductsMicroservice Kafka-

KafkaConsumerConfiguration.java

```
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.context.annotation.Configuration;
7 import org.springframework.core.env.Environment;
8 import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
9 import org.springframework.kafka.core.ConsumerFactory;
```

Problems Javadoc Declaration Console Progress

```
Consumer.invokeIfHaveRecords(KafkaMessageListenerContainer.java:1487) ~[spring-kafka-3.1.0.ja
Consumer.pollAndInvoke(KafkaMessageListenerContainer.java:1451) ~[spring-kafka-3.1.0.jar:3.1.
Consumer.run(KafkaMessageListenerContainer.java:1322) ~[spring-kafka-3.1.0.jar:3.1.0]
uture.java:1804) ~[na:na]

lled to deserialize
ationException(SerializationUtils.java:158) ~[spring-kafka-3.1.0.jar:3.1.0]
erialize(ErrorHandlingDeserializer.java:218) ~[spring-kafka-3.1.0.jar:3.1.0]
r.java:73) ~[kafka-clients-3.6.0.jar:na]
pletedFetch.java:300) ~[kafka-clients-3.6.0.jar:na]
mpletedFetch.java:263) ~[kafka-clients-3.6.0.jar:na]
tractFetch.java:340) ~[kafka-clients-3.6.0.jar:na]
tractFetch.java:306) ~[kafka-clients-3.6.0.jar:na]
er.java:1262) ~[kafka-clients-3.6.0.jar:na]
86) ~[kafka-clients-3.6.0.jar:na]
59) ~[kafka-clients-3.6.0.jar:na]
Consumer.pollConsumer(KafkaMessageListenerContainer.java:1658) ~[spring-kafka-3.1.0.jar:3.1.0
Consumer.doPoll(KafkaMessageListenerContainer.java:1633) ~[spring-kafka-3.1.0.jar:3.1.0]
Consumer.pollAndInvoke(KafkaMessageListenerContainer.java:1430) ~[spring-kafka-3.1.0.jar:3.1.

fault type provided
.jar:6.1.1]
JsonDeserializer.java:583) ~[spring-kafka-3.1.0.jar:3.1.0]
erialize(ErrorHandlingDeserializer.java:215) ~[spring-kafka-3.1.0.jar:3.1.0]

tedEventHandler    : Received a new event: iPhone 12
```

Boot Dashboard
- local
  - EmailNotificationMicroservice
  - ProductsMicroservice [:57953]

2 elements hidden by filter

And here's my email notification microservice, and I see that it was able to receive a new message,

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 41% Sun 25 Aug 2:25 AM

82. Trying how ErrorHandlingDeserializer works

Wed Dec 13 9:53 AM

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topi
c product-created-events-topic --property "parse.key=true" --property "key.separator=:"
>1:{sdfsdfsdfsdf}
>^C%
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topi
c product-created-events-topic --property "parse.key=true" --property "key.separator=:"
>2:dsafsdfsdf
```

I'll use message key and then anything that will cause a problem, let's send it.

1.25x 2:14 / 3:26

Udemy

82. Trying how ErrorHandlingDeserializer works

```
Caused by: org.springframework.kafka.support.serializer.DeserializationException: failed to d
        at org.springframework.kafka.support.serializer.SerializationUtils.deserializationExc
        at org.springframework.kafka.support.serializer.ErrorHandlingDeserializer.deserialize
        at org.apache.kafka.common.se
        at org.apache.kafka.clients.consumer.KafkaConsumer.pollForFetches(KafkaConsumer.java:
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1186) ~[ka
        at org.apache.kafka.clients.consumer.KafkaConsumer.poll(KafkaConsumer.java:1159) ~[ka
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        at org.springframework.kafka.listener.KafkaMessageListenerContainer$ListenerConsumer.
        ... 3 common frames omitted
Caused by: java.lang.IllegalStateException: No type information in headers and no default typ
        at org.springframework.util.Assert.state(Assert.java:76) ~[spring-core-6.1.1.jar:6.1.
        at org.springframework.kafka.support.serializer.JsonDeserializer.deserialize(JsonDese
        at org.springframework.kafka.support.serializer.ErrorHandlingDeserializer.deserialize
```

This exception is not in the loop and it is not continuously being thrown.

1.25x 2:34 / 3:26

POST http://localhost:57953/products

```
{
    "title":"iPhone 13",
    "price":800,
    "quantity":19
}
```

Status: 201 Created Time: 801 ms Size: 205 B

```
e5a19e31-6fdc-4808-9371-5a17494050ed
```

Now that I have created a new product, this means that a new product created event should be published

Chrome File Edit View History Bookmarks Profiles Tab Window Help 40% Sun 25 Aug 2:26 AM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Wed Dec 13 9:54 AM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

Package Explorer

- core
- EmailNotificationMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.emailnotification
      - EmailNotificationMicroserviceApplication.java
      - KafkaConsumerConfiguration.java
    - com.appsdeveloperblog.ws.emailnotification.handler
      - ProductCreatedEventHandler.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
    - target
    - HELP.md
    - mvnw
    - mvnw.cmd
    - pom.xml
- ProductsMicroservice [boot] [ProductsMicroservice Kafka-

Boot Dashboard

- local
  - EmailNotificationMicroservice
  - ProductsMicroservice [:57953]

2 elements hidden by filter

KafkaConsumerConfiguration.java

```
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.context.annotation.Configuration;
7 import org.springframework.core.env.Environment;
8 import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
9 import org.springframework.kafka.core.ConsumerFactory;
```

Problems Javadoc Declaration Console Progress

```
stenerConsumer.invokeIfHaveRecords(KafkaMessageListenerContainer.java:1487) ~[spring-kafka-3.
stenerConsumer.pollAndInvoke(KafkaMessageListenerContainer.java:1451) ~[spring-kafka-3.1.0.ja
stenerConsumer.run(KafkaMessageListenerContainer.java:1322) ~[spring-kafka-3.1.0.jar:3.1.0]
etableFuture.java:1804) ~[na:na]

on: failed to deserialize
erializationException(SerializationUtils.java:158) ~[spring-kafka-3.1.0.jar:3.1.0]
zer.deserialize(ErrorHandlingDeserializer.java:218) ~[spring-kafka-3.1.0.jar:3.1.0]
rializer.java:73) ~[kafka-clients-3.6.0.jar:na]
ord(CompletedFetch.java:300) ~[kafka-clients-3.6.0.jar:na]
ords(CompletedFetch.java:263) ~[kafka-clients-3.6.0.jar:na]
rds(AbstractFetch.java:340) ~[kafka-clients-3.6.0.jar:na]
tch(AbstractFetch.java:306) ~[kafka-clients-3.6.0.jar:na]
aConsumer.java:1262) ~[kafka-clients-3.6.0.jar:na]
java:1186) ~[kafka-clients-3.6.0.jar:na]
java:1159) ~[kafka-clients-3.6.0.jar:na]
stenerConsumer.pollConsumer(KafkaMessageListenerContainer.java:1658) ~[spring-kafka-3.1.0.jar
stenerConsumer.doPoll(KafkaMessageListenerContainer.java:1633) ~[spring-kafka-3.1.0.jar:3.1.0
stenerConsumer.pollAndInvoke(KafkaMessageListenerContainer.java:1430) ~[spring-kafka-3.1.0.ja

| no default type provided
-6.1.1.jar:6.1.1]
ialize(JsonDeserializer.java:583) ~[spring-kafka-3.1.0.jar:3.1.0]
zer.deserialize(ErrorHandlingDeserializer.java:215) ~[spring-kafka-3.1.0.jar:3.1.0]

uctCreatedEventHandler :
```

Udemy

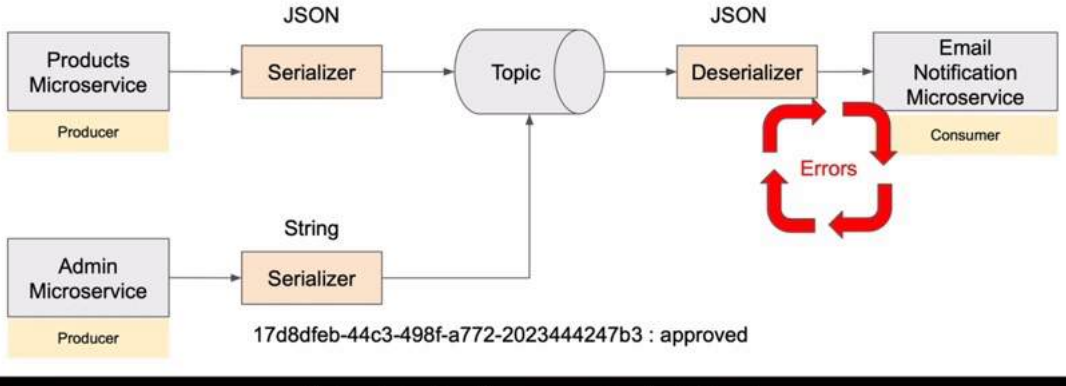
And here's a new message.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Sun 25 Aug 2:28 PM

84. Introduction to Dead Letter Topic(DLT) in Kafka

```
{
  "productId":"e9f6b0dd-19a9-4f8b-b958-98630564d04b",
  "title":"iPhone 5",
  "price":500,
  "quantity":19
}
```

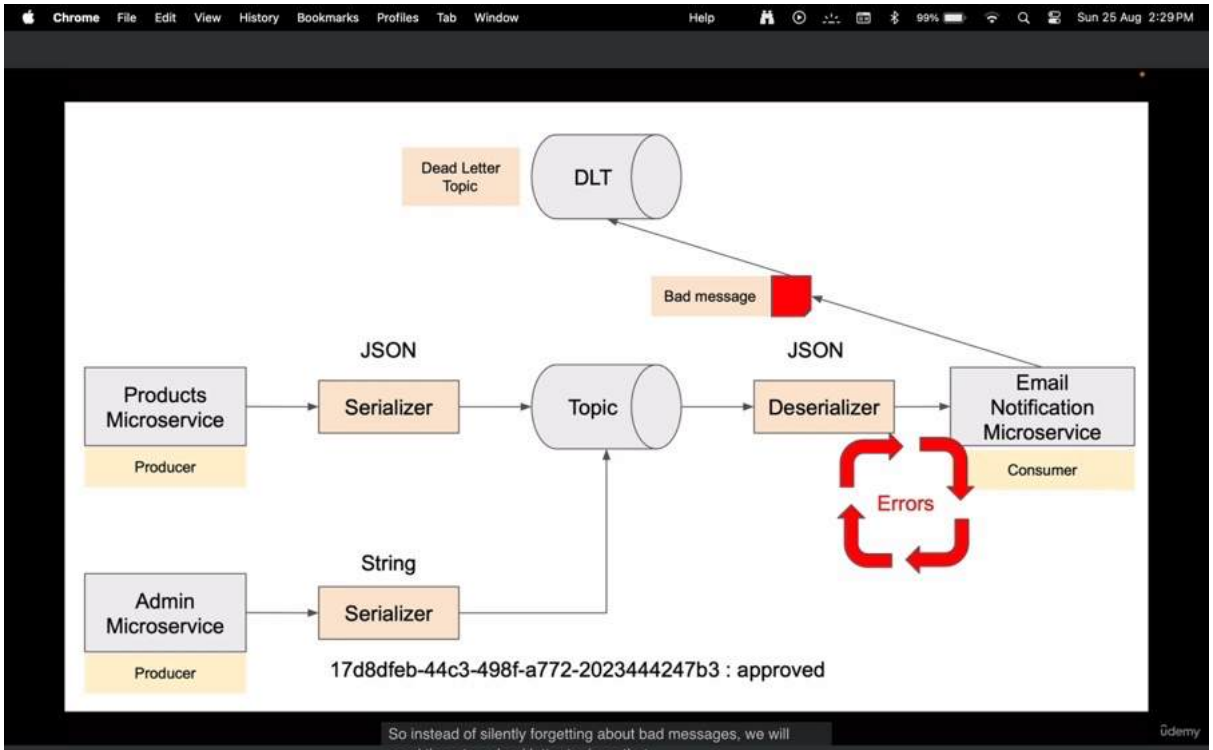
```
public class ProductCreatedEvent {
  private String productId;
  private String title;
  private BigDecimal price;
  private Integer quantity;
  ...
}
```

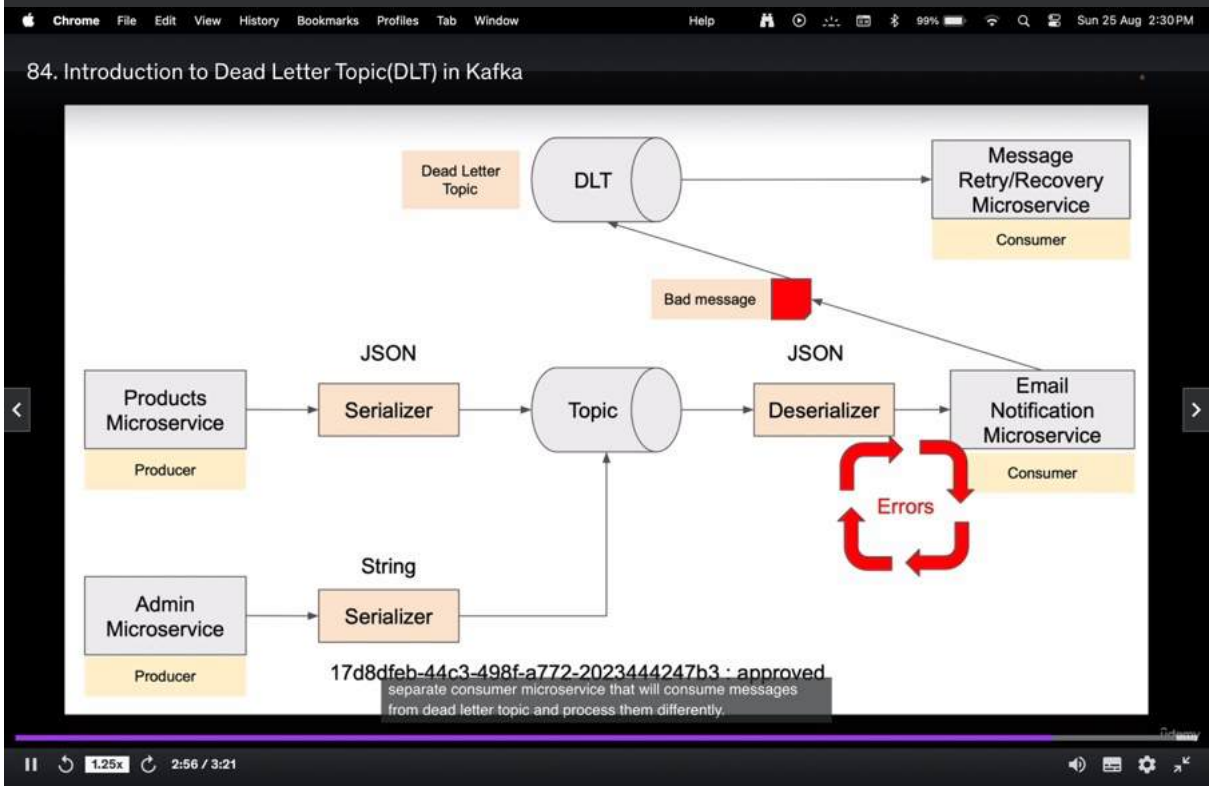


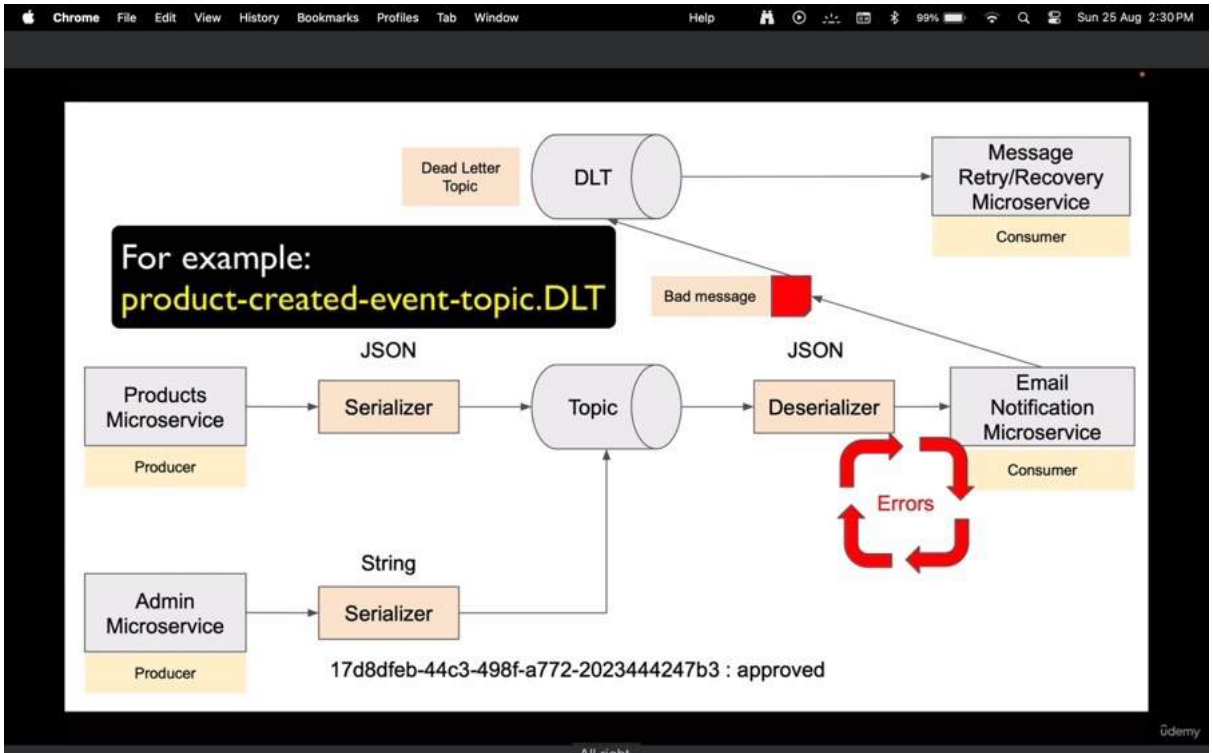
So in previous section of this course, we configured our Kafka consumer to recover from this situation

Udemy

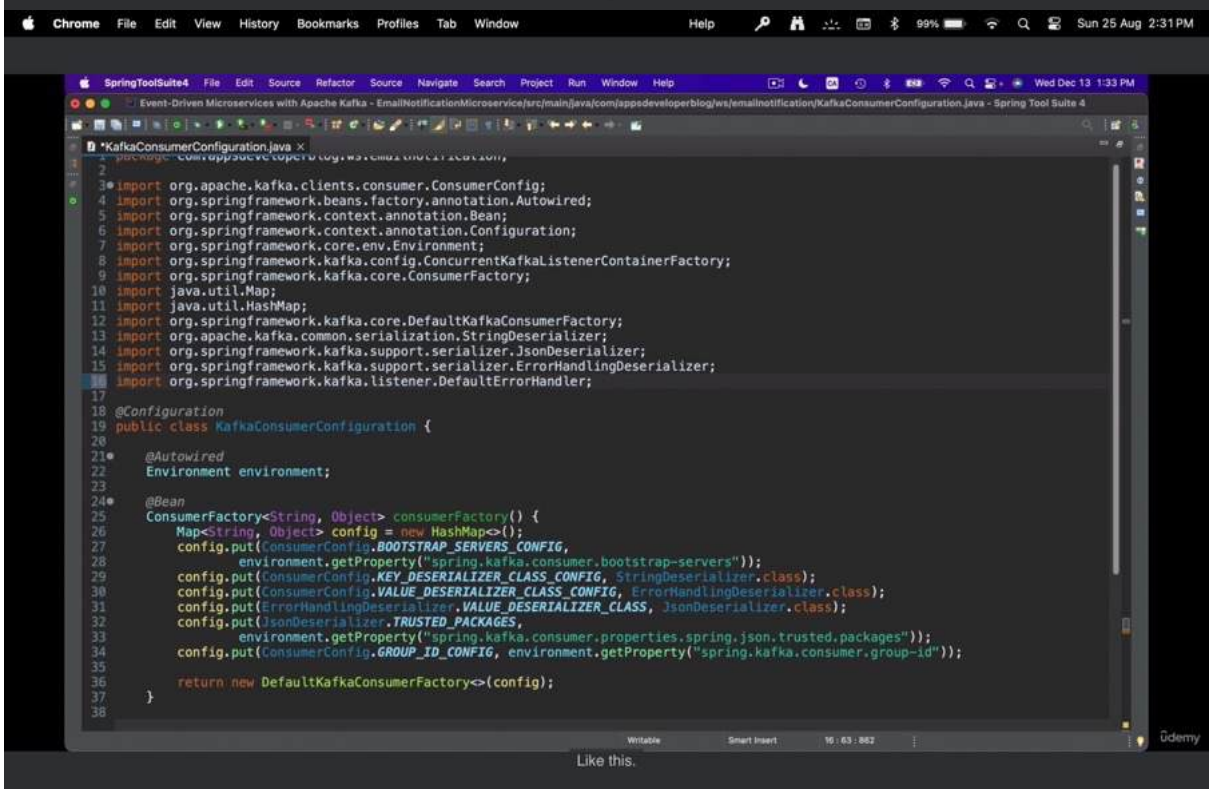
1.25x 1:44 / 3:21


Chrome File Edit View History Bookmarks Profiles Tab Window Help 99% Sun 25 Aug 2:29 PM
Dead Letter Topic
DLT
Bad message
JSON
JSON
Products Microservice
Producer
Serializer
Topic
Deserializer
Email Notification Microservice
Consumer
Errors
String
Admin Microservice
Producer
Serializer
17d8dfeb-44c3-498f-a772-2023444247b3 : approved
So instead of silently forgetting about bad messages, we will send them to a dead letter topic so that
Udemy


Chrome File Edit View History Bookmarks Profiles Tab Window Help 99% Sun 25 Aug 2:30 PM
84. Introduction to Dead Letter Topic(DLT) in Kafka
Dead Letter Topic
DLT
Message Retry/Recovery Microservice
Consumer
Bad message
JSON
JSON
Products Microservice
Producer
Serializer
Topic
Deserializer
Email Notification Microservice
Consumer
Errors
String
Admin Microservice
Producer
Serializer
17d8dfeb-44c3-498f-a772-2023444247b3 : approved
separate consumer microservice that will consume messages from dead letter topic and process them differently.
1.25x
2:56 / 3:21



All right.



```java
import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;
import org.springframework.kafka.listener.DefaultErrorHandler;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }
```

Like this.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;
import org.springframework.kafka.listener.DefaultErrorHandler;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler();

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        return factory;
    }
}
```

So this default error handler class, it provides error handling capabilities for Kafka consumers.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
import org.springframework.kafka.listener.DefaultErrorHandler;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }
}
```

And with this line of code here we say that use dead letter publishing recovery to publish failed messages

86. Create and Configure KafkaTemplate object

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

}
```

producer and provides methods to send messages to Kafka topics.

1.25x 2:10 / 6:08

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;
import org.springframework.kafka.listener.DeadLetterPublishingRecoverer;
import org.springframework.kafka.listener.DefaultErrorHandler;
import org.springframework.kafka.core.DefaultKafkaProducerFactory;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
```

Like this.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.apache.kafka.clients.consumer.ConsumerConfig;
import org.apache.kafka.clients.producer.ProducerConfig;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.kafka.config.ConcurrentKafkaListenerContainerFactory;
import org.springframework.kafka.core.ConsumerFactory;
import java.util.Map;
import java.util.HashMap;
import org.springframework.kafka.core.DefaultKafkaConsumerFactory;
import org.springframework.kafka.core.KafkaTemplate;
import org.springframework.kafka.core.ProducerFactory;
import org.apache.kafka.common.serialization.StringDeserializer;
import org.springframework.kafka.support.serializer.JsonDeserializer;
import org.springframework.kafka.support.serializer.ErrorHandlingDeserializer;
import org.springframework.kafka.listener.DeadLetterPublishingRecoverer;
import org.springframework.kafka.listener.DefaultErrorHandler;
import org.springframework.kafka.core.DefaultKafkaProducerFactory;
import org.springframework.kafka.support.serializer.JsonSerializer;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
```

Kafka support serializer package.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

*KafkaConsumerConfiguration.java

```java
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, JsonSerializer.class);
        config.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, StringSerializer.class);

        return new DefaultKafkaProducerFactory<>(config);
    }
```

Kafka producer will be created with this basic configuration.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 93% Sun 25 Aug 3:04 PM

Postman File Edit View Window Help Thu Dec 14 2:32 PM

Home Workspaces API Network Upgrade

POST http://localhost:51120

No Environment

http://localhost:51120/products

Save

POST http://localhost:51120/products Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "title":"iPhone 13",
3     "price":800,
4     "quantity":19
5 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 271 ms Size: 205 B Save as example

Pretty Raw Preview Visualize Text

```
1 25e4ead2-8b57-4fba-aa67-c47330e82208
```

So let's check if email notification microservice was able to receive this product created event.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 93% Sun 25 Aug 3:05 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Thu Dec 14 2:32 PM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

KafkaConsumerConfiguration.java

```
38          config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
39          config.put(JsonDeserializer.TRUSTED_PACKAGES,
40                  environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
41          config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));
42
43          return new DefaultKafkaConsumerFactory<>(config);
44      }
45
46      @Bean
47      ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
48              ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {
49
50          DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));
51
52          ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
53          factory.setConsumerFactory(consumerFactory);
54          factory.setCommonErrorHandler(errorHandler);
55
56          return factory;
57      }
58
59      @Bean
```

Problems Javadoc Declaration Console Progress

```
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-pr  t-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-pr  t-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-  uct-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-  uct-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consume  oduct-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consum  product-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=cons  product-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=cons  product-created-events-1, groupId=product-creat
[ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions assigned: [product-created-events-topic-
[ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received a new event: iPhone 13
```

And here I have a log message that confirms that email notification microservice was able to consume

```
Last login: Wed Dec 13 19:23:16 on ttys009
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --to
pic product-created-events-topic --property parse.key=true --property key.separator=:
>1:{sfasdsdf}
```

I will use one and then will provide Json that is not compatible with product created event Java class.

```
KafkaConsumerConfiguration.java
38          config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
39          config.put(JsonDeserializer.TRUSTED_PACKAGES,
40                  environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
41          config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));
42
43          return new DefaultKafkaConsumerFactory<>(config);
44      }
```

```
ients.producer.KafkaProducer      : [Producer clientId=producer-1] Instantiated an idempotent producer.
a.common.utils.AppInfoParser      : Kafka version: 3.6.0
a.common.utils.AppInfoParser      : Kafka commitId: 60e845626d8a465a
a.common.utils.AppInfoParser      : Kafka startTimeMs: 1702582411282
p.internals.TransactionManager    : [Producer clientId=producer-1] ProducerId set to 2003 with epoch 0
he.kafka.clients.NetworkClient    : [Producer clientId=producer-1] Error while fetching metadata with correlation id 2 : {product-created
he.kafka.clients.Metadata         : [Producer clientId=producer-1] Cluster ID: HvT9PAKcS5m8k2X5VrZjwg
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Revoke previous
KafkaMessageListenerContainer     : product-created-events: partitions revoked: [product-created-events-topic-0]
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining gr
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully jo
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assign
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully sy
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assig
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly as
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Found no commit
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Found no commit
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Found no commit
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Found no commit
c.internals.ConsumerCoordinator   : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset
c.internals.SubscriptionState     : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting offse
c.internals.SubscriptionState     : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting offse
KafkaMessageListenerContainer     : product-created-events: partitions assigned: [product-created-events-topic-0, product-created-events-
he.kafka.clients.NetworkClient    : [Consumer clientId=consumer-product-created-events-1, groupId=product-created-events] Node -1 disconn
```

This confirms that the error was handled and my Kafka consumer continues to listen for incoming events.

```
Last login: Thu Dec 14 14:29:45 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --to
pic product-created-events-topic.DLT --from-beginning --property print.key=true --property print.value=true
```

from which the message was originally read, but it will have the dot dot at the end.

```
Last login: Thu Dec 14 14:29:45 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --to
pic product-created-events-topic.DLT --from-beginning --property print.key=true --property print.value=true
1       "e3NmYXNkc2RmfQ==="
```

The message looks different here, and this is because this is actually a base64 encoded version of

Chrome File Edit View History Bookmarks Profiles Tab Window Help 92% Sun 25 Aug 3:07 PM

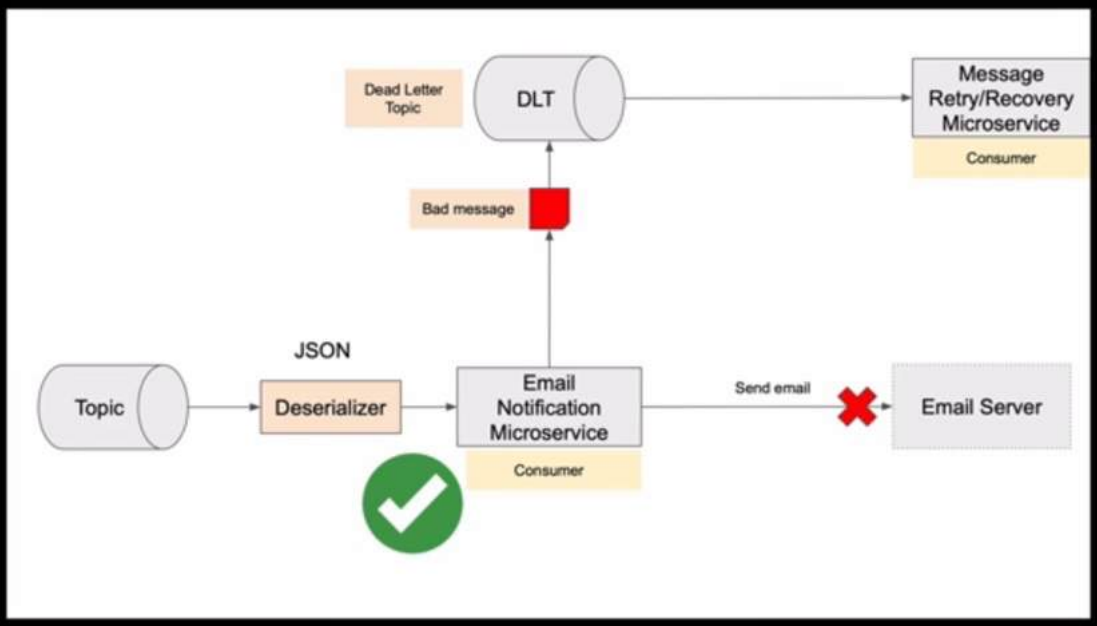
< DECODE > Decodes your data into the area below.

{sfasdsdf}

Copy to clipboard

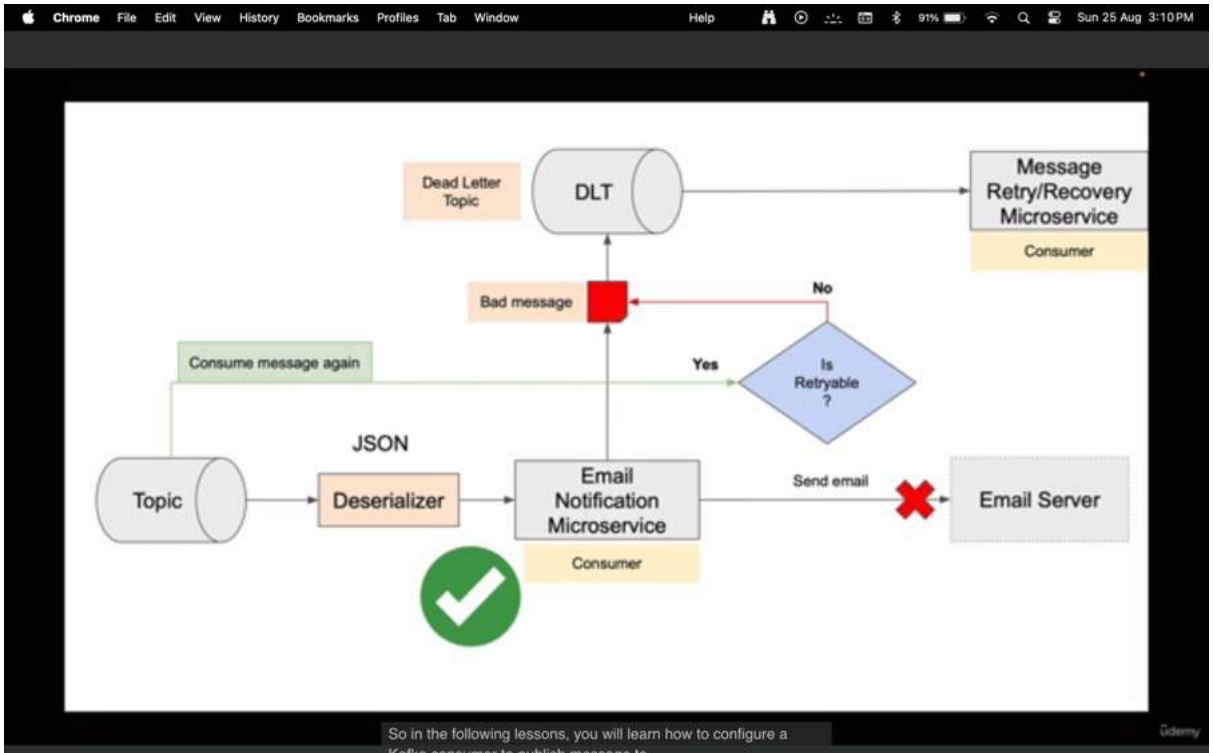
Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 91% Sun 25 Aug 3:09 PM

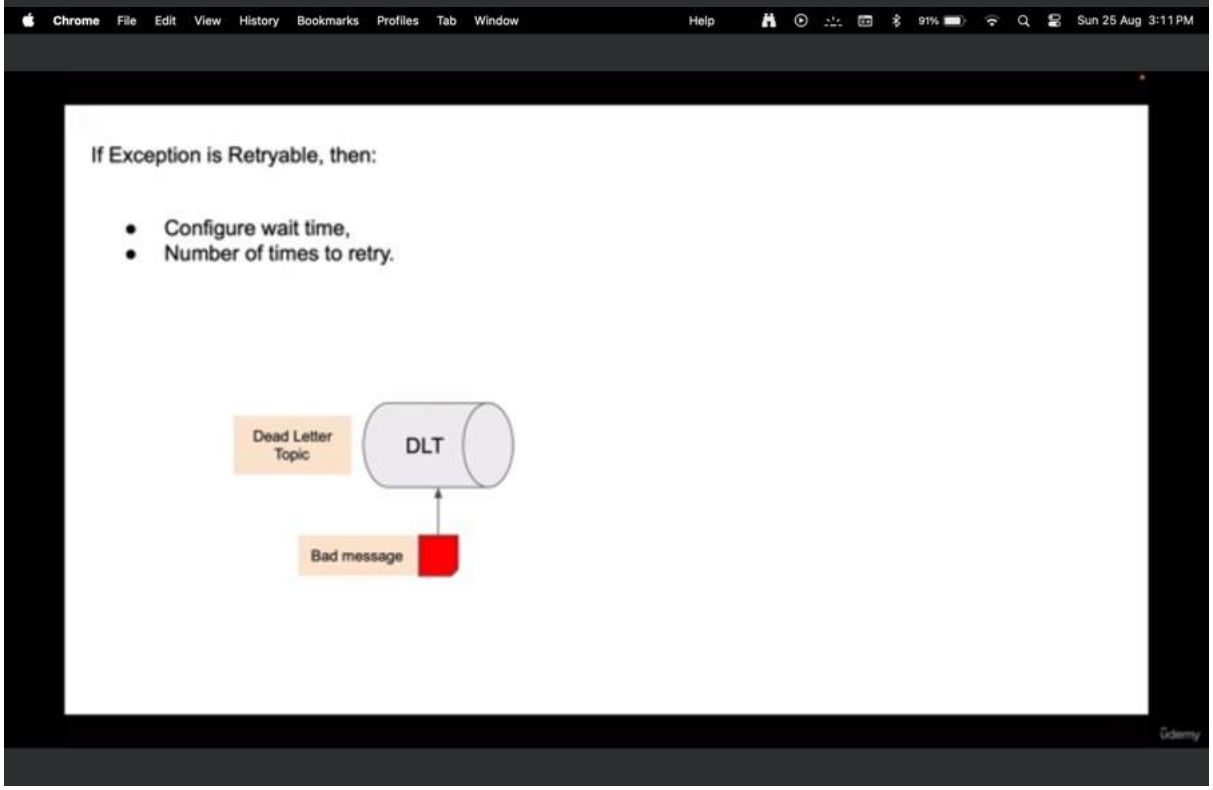


Now notice that in this case, email notification microservice or Kafka consumer did consume message

Udemy



So in the following lessons, you will learn how to configure a Kafka consumer to publish message to

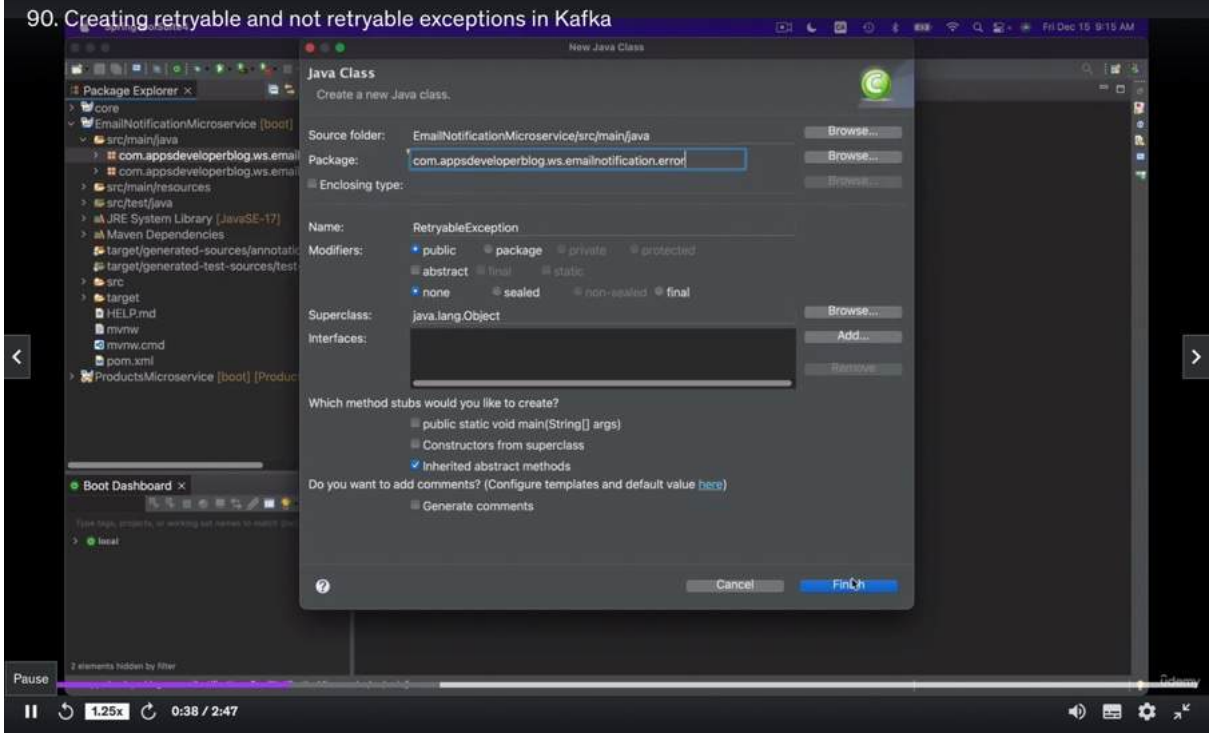


If Exception is Retryable, then:

- Configure wait time,
- Number of times to retry.

And in the following lessons you will learn how to handle retrieval and not retrieval exceptions in

90. Creating retryable and not retryable exceptions in Kafka



90. Creating retryable and not retryable exceptions in Kafka

Generate Constructors from Superclass

Select constructors to implement:

- RuntimeException()
- RuntimeException(String, Throwable, boolean, boolean)
- RuntimeException(String, Throwable)
- [x] RuntimeException(String)
- [x] RuntimeException(Throwable)

Select All

Deselect All

I will add a constructor that accepts custom message, and I will add a constructor that accepts throwable.

Insertion point:

Pause

1.25x 1:13 / 2:47

90. Creating retryable and not retryable exceptions in Kafka

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/error/RetryableException.java - Spring Tool Suite 4

Package Explorer

- core
- EmailNotificationMicroservice [boot]
  - src/main/java
    - com.appsdeveloperblog.ws.emailnotificatio
    - com.appsdeveloperblog.ws.emailnotificatio
      - RetryableException.java
    - com.appsdeveloperblog.ws.emailnotificatio
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - target/generated-sources/annotations
  - target/generated-test-sources/test-annotatio
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
- ProductsMicroservice [boot] [ProductsMicrose

Boot Dashboard

RetryableException.java

```
1  package com.appsdeveloperblog.ws.emailnotification.error;
2
3  public class RetryableException extends RuntimeException {
4
5      public RetryableException(String message) {
6          super(message);
7          // TODO Auto-generated constructor stub
8      }
9
10     public RetryableException(Throwable cause) {
11         super(cause);
12         // TODO Auto-generated constructor stub
13     }
14
15
16
17 }
18
```

And let's add one more that we'll use to throw a Non-writable exception.

1.25x 1:45 / 2:47

90. Creating retryable and not retryable exceptions in Kafka

Java Class

Create a new Java class.

Source folder: EmailNotificationMicroservice/src/main/java

Package: com.appsdeveloperblog.ws.emailnotification.error

Name: NotRetryableException

Modifiers: public package private protected abstract final static none sealed non-sealed final

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

Generate comments

Cancel Finish

And click on finish button to create this class.

2:06 / 2:47

90. Creating retryable and not retryable exceptions in Kafka

```
package com.appsdeveloperblog.ws.emailnotification.error;

public class NotRetryableException extends RuntimeException {

    public NotRetryableException(String message) {
        super(message);
        // TODO Auto-generated constructor stub
    }

    public NotRetryableException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

}
```

So now that our custom exception classes are ready, let's save changes to not retry exception.

2:38 / 2:47

91. Configure DefaultErrorHandler with a list of not retryable exceptions

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

RetryableException.java | NotRetryableException.java | *KafkaConsumerConfiguration.java

```
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));
        errorHandler.addNotRetryableExceptions(NotRetryableException.class, HttpServerErrorException.class);

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, JsonSerializer.class);
        config.put(ProducerConfig.
```

Or instead of registering multiple exceptions here in this list, you can catch them in your Kafka listener

1.25x 2:11 / 2:45

91. Configure DefaultErrorHandler with a list of not retryable exceptions

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

RetryableException.java | NotRetryableException.java | *KafkaConsumerConfiguration.java

```
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate));
        errorHandler.addNotRetryableExceptions(NotRetryableException.class);

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, JsonSerializer.class);
        config.put(ProducerConfig.
```

So now that I have configured default error handler with my own not retrievable exception, let's save

1.25x 2:31 / 2:45

92. Trying how not retryable exception works

RetryableException.java | NotRetryableException.java | KafkaConsumerConfiguration.java | *ProductCreatedEventHandler.java

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        if(true) throw new NotRetryableException("An error took place. No need to consume this message again.");
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
    }

}
```

1.25x 1:19 / 6:24

92. Trying how not retryable exception works

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topi
c product-created-events-topic.DLT --from-beginning --property print.key=true --property print.value=true
1       "e3NmYXNkc2RmfQ=="
```

1.25x 3:41 / 6:24

92. Trying how not retryable exception works

```
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics="product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
```

```
        ssl.keystore.certificate.chain = null
        ssl.keystore.key = null
        ssl.keystore.location = null
        ssl.keystore.password = null
        ssl.keystore.type = JKS
        ssl.protocol = TLSv1.3
        ssl.provider = null
        ssl.secure.random.implementation = null
        ssl.trustmanager.algorithm = PKIX
        ssl.truststore.certificates = null
        ssl.truststore.location = null
        ssl.truststore.password = null
        ssl.truststore.type = JKS
        transaction.timeout.ms = 60000
        transactional.id = null
        value.serializer = class org.springframework.kafka.support.serializer.JsonSerializer

2023-12-15T12:13:12.849-05:00  INFO 50346 --- [ntainer#0-0-C-1] o.a.k.clients.producer.KafkaProducer
2023-12-15T12:13:12.860-05:00  INFO 50346 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser
2023-12-15T12:13:12.860-05:00  INFO 50346 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser
2023-12-15T12:13:12.860-05:00  INFO 50346 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser
2023-12-15T12:13:12.866-05:00  INFO 50346 --- [ad | producer-1] org.apache.kafka.clients.Metadata
2023-12-15T12:13:12.876-05:00  INFO 50346 --- [ad | producer-1] o.a.k.c.p.internals.TransactionManager
```

```
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic product-created-events-topic.DLT --from-beginning --property print.key=true --property print.value=true
1       "e3NmYXNkc2RmfQ=="
7472fded-cac2-45c8-adb9-b5d82d0c00fe    {"productId":"7472fded-cac2-45c8-adb9-b5d82d0c00fe","title":"iPhone 13","price":800,"quantity":19}
```

93. Register RetryableException and define wait time interval

```
    config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spr

    return new DefaultKafkaConsumerFactory<>(config);
}

@Bean
ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerConta
        ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<Str

    DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLett
    errorHandler.addNotRetryableExceptions(NotRetryableException.class);
    errorHandler.addRetryableExceptions(RetryableException.class);

    ConcurrentKafkaListenerContainerFactory<String, Object> factory = new (
    factory.setConsumerFactory(consumerFactory);
    factory.setCommonErrorHandler(errorHandler);

    return factory;
}

@Bean
KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object>
    return new KafkaTemplate<>(producerFactory);
}
```

1.25x 1:01 / 2:27

93. Register RetryableException and define wait time interval

```
    return new DefaultKafkaConsumerFactory<>(config);
}

@Bean
ConcurrentKafkaListenerContainerFactory<String, Object> kafkal
        ConsumerFactory<String, Object> consumerFactory, Kafka

    DefaultErrorHandler errorHandler = new DefaultErrorHandle
            new FixedBackOff(5000,3));
    errorHandler.addNotRetryableExceptions(NotRetryableExcept
    errorHandler.addRetryableExceptions(RetryableException.cl

    ConcurrentKafkaListenerContainerFactory<String, Object> fa
    factory.setConsumerFactory(consumerFactory);
    factory.setCommonErrorHandler(errorHandler);

    return factory;
}

@Bean
```

1.25x 1:44 / 2:27

94. Throwing a RetryableException

```java
package com.appsdeveloperblog.ws.emailnotification;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class EmailNotificationMicroserviceApplication {

    public static void main(String[] args) {
        SpringApplication.run(EmailNotificationMicroserviceApplication.class, args);
    }

    @Bean
    RestTemplate getRestTemplate() {
        return new RestTemplate();
    }

}
```

So let's go to product created event handler.

2:21 / 7:49

94. Throwing a RetryableException

```java
import com.appsdeveloperblog.ws.core.ProductCreatedEvent;
import com.appsdeveloperblog.ws.emailnotification.error.NotRetryableException;
import com.appsdeveloperblog.ws.emailnotification.error.RetryableException;

@Component
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());

        String requestUrl = "http://localhost:8082";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        }

    }

}
```

This version of Spring Tool suite is very glitchy for me.

6:11 / 7:49

94. Throwing a RetryableException

*ProductCreatedEventHandler.java × EmailNotificationMicroserviceApplication.java RetryableException.java

```
19 @Component
20 @KafkaListener(topics = "product-created-events-topic")
21 public class ProductCreatedEventHandler {
22
23     private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
24     private RestTemplate restTemplate;
25
26     public ProductCreatedEventHandler(RestTemplate restTemplate) {
27         this.restTemplate = restTemplate;
28     }
29
30     @KafkaHandler
31     public void handle(ProductCreatedEvent productCreatedEvent) {
32         LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
33
34         String requestUrl = "http://localhost:8082";
35
36         try {
37             ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);
38
39             if (response.getStatusCode().value() == HttpStatus.OK.value()) {
40                 LOGGER.info("Received response from a remote service: " + response.getBody());
41             }
42         } catch (ResourceAccessException ex) {
43             LOGGER.error(ex.getMessage());
44             throw new RetryableException(ex);
45         } catch(HttpServerErrorException ex) {
46             LOGGER.error(ex.getMessage());
47             throw new NotRetryableException(ex);
48         } catch(Exception ex) {
49             LOGGER.error(ex.getMessage());
50             throw new NotRetryableException(ex);
51         }
52
53     }
54
55 }
```

So if remote microservice is not available we will throw retry exception.

Pause 1.25x 7:15 / 7:49

94. Throwing a RetryableException

ProductCreatedEventHandler.java × EmailNotificationMicroserviceApplication.java RetryableException.java

```
19 @Component
20 @KafkaListener(topics = "product-created-events-topic")
21 public class ProductCreatedEventHandler {
22
23     private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
24     private RestTemplate restTemplate;
25
26     public ProductCreatedEventHandler(RestTemplate restTemplate) {
27         this.restTemplate = restTemplate;
28     }
29
30     @KafkaHandler
31     public void handle(ProductCreatedEvent productCreatedEvent) {
32         LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
33
34         String requestUrl = "http://localhost:8082";
35
36         try {
37             ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);
38
39             if (response.getStatusCode().value() == HttpStatus.OK.value()) {
40                 LOGGER.info("Received response from a remote service: " + response.getBody());
41             }
42         } catch (ResourceAccessException ex) {
43             LOGGER.error(ex.getMessage());
44             throw new RetryableException(ex);
45         } catch(HttpServerErrorException ex) {
46             LOGGER.error(ex.getMessage());
47             throw new NotRetryableException(ex);
48         } catch(Exception ex) {
49             LOGGER.error(ex.getMessage());
50             throw new NotRetryableException(ex);
51         }
52
53     }
54
55 }
```

And now we need a microservice to send this Http request to.

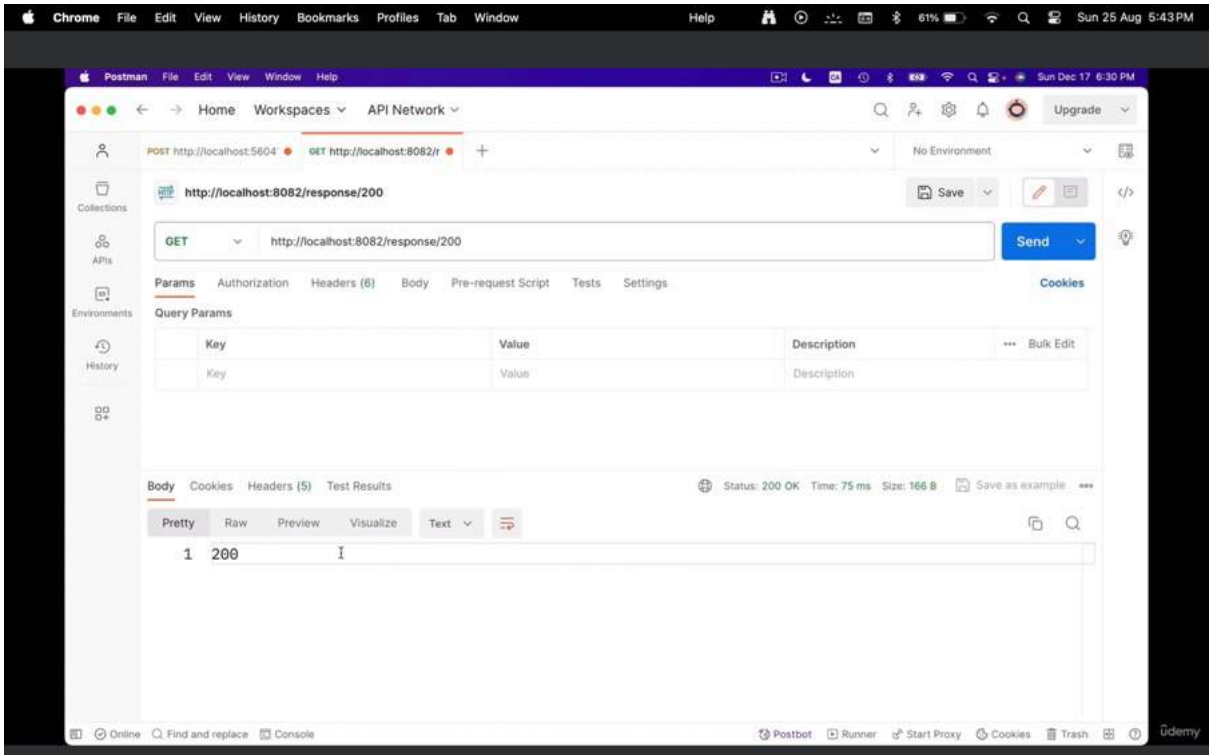
Pause 1.25x 7:42 / 7:49

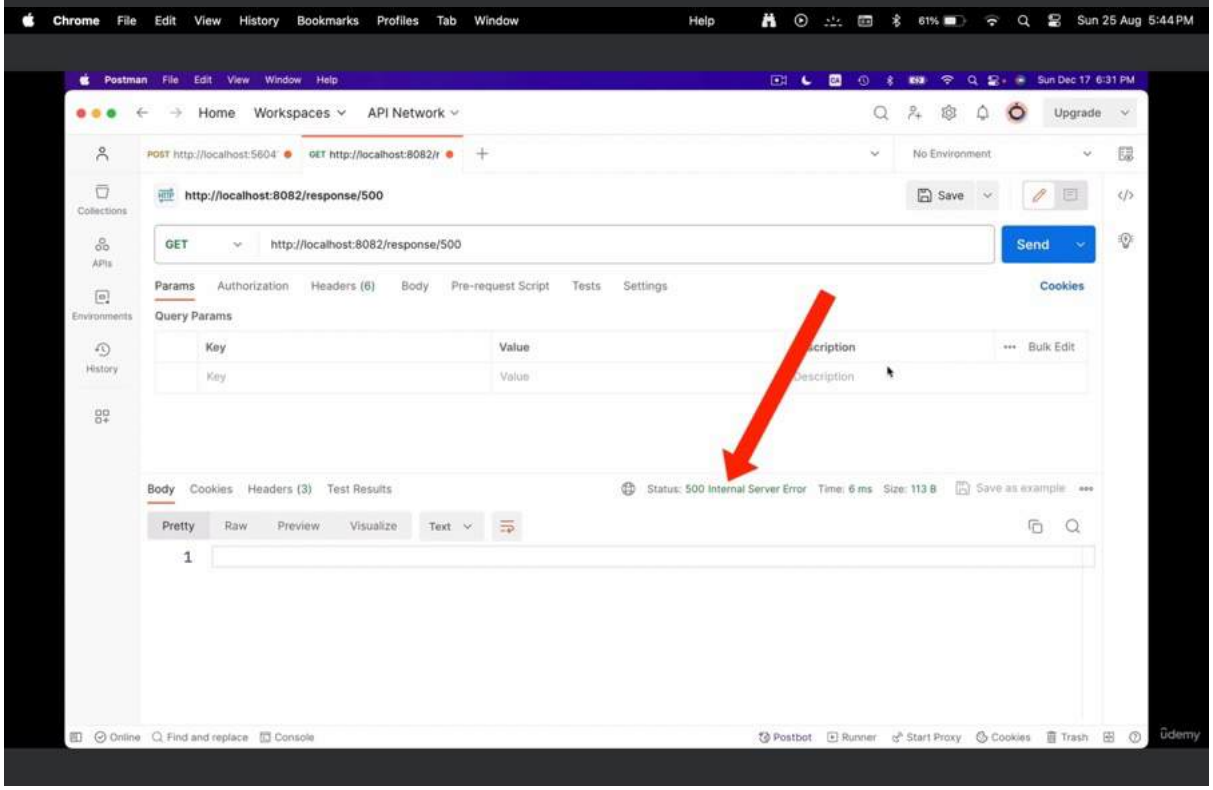
Event-Driven Microservices with Apache Kafka - mockservice/src/main/java/com/appsdeveloperblog/ws/mockservice/StatusCheckController.java - Spring Tool Suite 4

```java
package com.appsdeveloperblog.ws.mockservice;

import org.springframework.http.ResponseEntity;

@RestController
@RequestMapping("/response")
public class StatusCheckController {

    @GetMapping("/200")
    ResponseEntity<String> response200String() {
        return ResponseEntity.ok().body("200");
    }

    @GetMapping("/500")
    ResponseEntity<String> response500String() {
        return ResponseEntity.internalServerError().build();
    }

}
```

Event-Driven Microservices with Apache Kafka - mockservice/src/main/resources/application.properties - Spring Tool Suite 4

```properties
server.port=8082
```


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 5:43 PM
Postman File Edit View Window Help
Sun Dec 17 6:30 PM
Home Workspaces API Network
Upgrade
POST http://localhost:5604
GET http://localhost:8082/r
No Environment
Collections
APIs
Environments
History
http://localhost:8082/response/200
Save
GET
http://localhost:8082/response/200
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body Cookies Headers (5) Test Results
Status: 200 OK Time: 75 ms Size: 166 B
Save as example
Pretty Raw Preview Visualize Text
1 200
Online Find and replace Console
Postbot Runner Start Proxy Cookies Trash
Udemy


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 5:44 PM
Postman File Edit View Window Help
Sun Dec 17 6:31 PM
Home Workspaces API Network
Upgrade
POST http://localhost:5604
GET http://localhost:8082/r
No Environment
Collections
APIs
Environments
History
http://localhost:8082/response/500
Save
GET
http://localhost:8082/response/500
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
Query Params
Key
Value
Bulk Edit
Body Cookies Headers (3) Test Results
Status: 500 Internal Server Error Time: 6 ms Size: 113 B
Save as example
Pretty Raw Preview Visualize Text
1
Online Find and replace Console
Postbot Runner Start Proxy Cookies Trash
Udemy

```java
19 @Component
20 @KafkaListener(topics = "product-created-events-topic")
21 public class ProductCreatedEventHandler {
22
23     private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
24     private RestTemplate restTemplate;
25
26     public ProductCreatedEventHandler(RestTemplate restTemplate) {
27         this.restTemplate = restTemplate;
28     }
29
30     @KafkaHandler
31     public void handle(ProductCreatedEvent productCreatedEvent) {
32         LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
33
34         String requestUrl = "http://localhost:8082/response/200";
35
36         try {
37             ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);
38
39             if (response.getStatusCode().value() == HttpStatus.OK.value()) {
40                 LOGGER.info("Received response from a remote service: " + response.getBody());
41             }
42         } catch (ResourceAccessException ex) {
43             LOGGER.error(ex.getMessage());
44             throw new RetryableException(ex);
45         } catch(HttpServerErrorException ex) {
46             LOGGER.error(ex.getMessage());
47             throw new NotRetryableException(ex);
48         } catch(Exception ex) {
49             LOGGER.error(ex.getMessage());
50             throw new NotRetryableException(ex);
51         }
52
53     }
54
55 }
56
```

But before starting this application, I will also set a couple of debugger breakpoints.

96. Trying how retry works

Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

Remember my decision

No | Switch

```
2023-12-18T09:27:06.753-05:00  INFO 3857 --- [main] c.a.w.m.MockserviceApplication
2023-12-18T09:27:07.287-05:00  INFO 3857 --- [main] o.s.b.w.embedded.tomcat.TomcatW
2023-12-18T09:27:07.294-05:00  INFO 3857 --- [main] o.apache.catalina.core.Standard
2023-12-18T09:27:07.294-05:00  INFO 3857 --- [main] o.apache.catalina.core.Standard
2023-12-18T09:27:07.347-05:00  INFO 3857 --- [main] o.a.c.c.C.[Tomcat].[localhost].
2023-12-18T09:27:07.347-05:00  INFO 3857 --- [main] w.s.c.ServletWebServerApplicat
2023-12-18T09:27:07.555-05:00  INFO 3857 --- [main] o.s.b.w.embedded.tomcat.TomcatW
2023-12-18T09:27:07.561-05:00  INFO 3857 --- [main] c.a.w.m.MockserviceApplication
2023-12-18T09:31:23.810-05:00  INFO 3857 --- [nio-8082-exec-1] o.a.c.c.C.[Tomcat].[localhost]
2023-12-18T09:31:23.810-05:00  INFO 3857 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServl
2023-12-18T09:31:23.811-05:00  INFO 3857 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServl
```

1.25x 3:38 / 8:13

96. Trying how retry works

Chrome File Edit View History Bookmarks Profiles Tab Window Help 57% Sun 25 Aug 6:00 PM

Home Workspaces API Network Upgrade

POST http://localhost:4963 GET http://localhost:8082/r No Environment

http://localhost:49630/products Save

POST http://localhost:49630/products Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "title":"iPad Pro",
    "price":1200,
    "quantity":19
}
```

Response

Could not send request

Error: connect ECONNREFUSED 127.0.0.1:56041 | View in Console

Let's send it again.

1.25x 3:36 / 8:13

96. Trying how retry works

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

ProductCreatedEventHandler.java

```
@Component
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = re  mplate;
    }

    @KafkaHandler
    public void handle(ProductC  edEvent productCreatedEvent) {
        LOGGER.info("Received a  event: " + productCreatedEvent.getTitle());

        String requestUrl = "htt  /localhost:8082/response/200";

        try {
            ResponseEntity<Str  sponse = restTemplate.exchange(requestUrl, HttpMethod.GET, nu

            if (response.getStatus ode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch(HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch(Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

    }
}
```

On the line that says that I have received a successful response from Mog service.

1.25x 3:45 / 8:13

96. Trying how retry works

```
19 @Component
20 @KafkaListener(topics = "product-created-events-topic")
21 public class ProductCreatedEventHandler {
22
23     private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
24     private RestTemplate restTemplate;
25
26     public ProductCreatedEventHandler(RestTemplate restTemplate) {
27         this.restTemplate = restTemplate;
28     }
29
```

```
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-product-created-eve
o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions assigned: [p
c.a.w.e.h.ProductCreatedEventHandler     : Received a new event: iPad Pro
c.a.w.e.h.ProductCreatedEventHandler     : Received response from a remote service: 200
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions revoked: [pr
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.ConsumerCoordinator  : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-product-created-eve
o.a.k.c.c.internals.SubscriptionState    : [Consumer clientId=consumer-product-created-eve
o.s.k.l.KafkaMessageListenerContainer    : product-created-events: partitions assigned: [p
```

1.25x 3:59 / 8:13

96. Trying how retry works

```
19 @Component
20 @KafkaListener(topics = "product-created-events-topic")
21 public class ProductCreatedEventHandler {
22
23     private final Logger LOGGER = LoggerFactory.getLogger(this.getClass())
24     private RestTemplate restTemplate;
25
26     public ProductCreatedEventHandler(RestTemplate restTemplate) {
27         this.restTemplate = restTemplate;
28     }
29
```

Console | Problems | Debug Shell | Boot Dashboard

local
- EmailNotificationMicroservice [:49574]
- mockservice [:8082]
- ProductsMicroservice [:49630]

Actually, I will need to open up Spring Boot dashboard, select Mock Service and then click on this

1.25x 4:28 / 8:13

96. Trying how retry works

```
POST http://localhost:49630/products
{
    "title":"iPad Pro",
    "price":1200,
    "quantity":19
}
```

```
cba195c2-6d7e-4a38-8429-a6a4839e5c0a
```

1.25x 4:41 / 8:13

96. Trying how retry works

```
try {
    ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, nu
    if (response.getStatusCode().value() == HttpStatus.OK.value()) {
        a remote service: " + response.getBody());
    }
} catch (ResourceAccessException ex) {
    LOGGER.error(ex.getMessage());
    throw new RetryableException(ex);
} catch(HttpServerErrorException ex) {
    throw new NotRetryableException(ex);
} catch(Exception ex) {
    LOGGER.error(ex.getMessage());
    throw new NotRetryableException(ex);
}
```

```
2023-12-18T09:26:45.441-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionState
2023-12-18T09:26:45.460-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer
2023-12-18T09:31:23.727-05:00  INFO 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
2023-12-18T09:31:41.101-05:00  INFO 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
2023-12-18T09:31:41.108-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.109-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.109-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer
2023-12-18T09:31:41.109-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.113-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.114-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.115-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.116-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.116-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.117-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
2023-12-18T09:31:41.117-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator
```

2023-1 And this time I get a debugger breakpoint triggered on the line
that catches resource access exception.

1.25x 4:48 / 8:13

96. Trying how retry works

```
ProductCreatedEventHandler.java
        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, nu

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch(HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch(Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }
```

```
Console
        at java.base/sun.net.www.http.HttpClient.openServer(HttpClient.java:532) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.openServer(HttpClient.java:637) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.<init>(HttpClient.java:280) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.New(HttpClient.java:385) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.New(HttpClient.java:407) ~[na:na]
        at java.base/sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(HttpURLConnection.jav
        at java.base/sun.net.www.protocol.http.HttpURLConnection.plainConnect0(HttpURLConnection.java:1
        at java.base/sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:11
        at java.base/sun.net.www.protocol.http.HttpURLConnection.connect(HttpURLConnection.java:1057) ~
        at org.springframework.http.client.SimpleClientHttpRequest.executeInternal(SimpleClientHttpRequ
        at org.springframework.http.client.AbstractStreamingClientHttpRequest.executeInternal(AbstractS
        at org.springframework.http.client.AbstractClientHttpRequest.execute(AbstractClientHttpRequest.
        at org.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:879) ~[spring-web-6.
        ... 27 common frames omitted

2023-12-18T09:34:08.965-05:00  INFO 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
2023-12-18T09:34:08.967-05:00 ERROR 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
```

Let's resume it.

1.25x 5:03 / 8:13

96. Trying how retry works

```
ProductCreatedEventHandler.java   KafkaConsumerConfiguration.java
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaT
                new FixedBackOff(5000,3));

        errorHandler.addRetryableExceptions(RetryableException.class);

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContai
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }
```

```
Console
        at java.base/sun.net.www.http.HttpClient.openServer(HttpClient.java:637) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.<init>(HttpClient.java:280) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.New(HttpClient.java:385) ~[na:na]
        at java.base/sun.net.www.http.HttpClient.New(HttpClient.java:407) ~[na:na]
        at java                                                    ttpClient(HttpURLConnection.java:1309) ~
        at java                                                    nect0(HttpURLConnection.java:1242) ~[na
        at java                                                    nect(HttpURLConnection.java:1128) ~[na:
```

Here we've created the default handler object, and we configured it with the fixed backoff.

1.25x 5:59 / 8:13

96. Trying how retry works

```
Last login: Mon Dec 18 09:17:47 on ttys002
sergeykargopolov@Sergeys-MacBook-Pro kafka % ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic
product-created-events-topic.DLT --property print.key=true --property print.value=true
[2023-12-18 09:29:36,495] WARN [Consumer clientId=console-consumer, groupId=console-consumer-76235] Error while fetchin
g metadata with correlation id 2 : {product-created-events-topic.DLT=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clie
nts.NetworkClient)
[2023-12-18 09:29:36,602] WARN [Consumer clientId=console-consumer, groupId=console-consumer-76235] Error while fetchin
g metadata with correlation id 7 : {product-created-events-topic.DLT=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clie
nts.NetworkClient)
[2023-12-18 09:29:36,704] WARN [Consumer clientId=console-consumer, groupId=console-consumer-76235] Error while fetchin
g metadata with correlation id 8 : {product-created-events-topic.DLT=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clie
nts.NetworkClient)
[2023-12-18 09:29:36,806] WARN [Consumer clientId=console-consumer, groupId=console-consumer-76235] Error while fetchin
g metadata with correlation id 9 : {product-created-events-topic.DLT=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.clie
nts.NetworkClient)
[2023-12-18 09:29:36,907] WARN [Consumer clientId=console-consumer, groupId=console-consumer-76235] Error while fetchin
g metadata with correlation id 10 : {product-created-events-topic.DLT=UNKNOWN_TOPIC_OR_PARTITION} (org.apache.kafka.cli
ents.NetworkClient)
14d80a50-234c-4b6c-a718-c58a3fe542a8    {"productId":"14d80a50-234c-4b6c-a718-c58a3fe542a8","title":"iPad Pro","price":
1200,"quantity":19}
```

I'll go to a terminal window and here is our message in the deadlier topic.

1.25x 6:48 / 8:13

96. Trying how retry works

POST http://localhost:49630/products

Body:
```
{
    "title":"iPad Pro",
    "price":1250,
    "quantity":19
}
```

Status: 201 Created Time: 26 ms Size: 205 B

Response:
```
14d80a50-234c-4b6c-a718-c58a3fe542a8
```

1.25x 7:03 / 8:13

```
@KafkaHandler
public void handle(ProductCreatedEvent productCreatedEvent) {
    LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());

    String requestUrl = "http://localhost:8082/response/200";

    try {
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, Strin

        if (response.getStatusCode().value() == HttpStatus.OK.value()) {
            LOGGER.info("Received response from a remote service: " + response.getBody());

    } catch (ResourceAccessException ex) {
        LOGGER.error(ex.getMessage());
        throw new RetryableException(ex);

        LOGGER.error(ex.getMessage());
        throw new NotRetryableException(ex);
    } catch(Exception ex) {
        LOGGER.error(ex.getMessage());
        throw new NotRetryableException(ex);
    }

}
}
```

```
2023-12-18T09:37:48.484-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
2023-12-18T09:37:48.484-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
2023-12-18T09:37:48.489-05:00  INFO 3851 --- [ad | producer-1] org.apache.kafka.clients.Metadata
2023-12-18T09:37:48.489-05:00  INFO 3851 --- [ad | producer-1] o.a.k.c.p.internals.TransactionManager
2023-12-18T09:38:44.769-05:00  INFO 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
2023-12-18T09:38:44.772-05:00 ERROR 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler
```

The destination microservice is still not available.

```
@KafkaHandler
public void handle(ProductCreatedEvent productCreatedEvent) {
    LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());

    String requestUrl = "http://localhost:8082/response/200";

    try {
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, Strin

        if (response.getStatusCode().value() == HttpStatus.OK.value()) {
            LOGGER.info("Received response from a remote service: " + response.getBody());
        }
    } catch (ResourceAccessException ex) {
        LOGGER.error(ex.getMessage());
        throw new RetryableException(ex);
    } catch(HttpServerErrorException ex) {
        LOGGER.error(ex.getMessage());
        throw new NotRetryableException(ex);
    } catch(Exception ex) {
        LOGGER.error(ex.getMessage());
        throw new NotRetryableException(ex);
    }
```

```
3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
3851 --- [ad | producer-1] org.apache.kafka.clients.Metadata        : [Produ
3851 --- [ad | producer-1] o.a.k.c.p.internals.TransactionManager   : [Produ
3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Receiv
3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : I/O er
```

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

ProductCreatedEventHandler.java × KafkaConsumerConfiguration.java

```java
29
30●     @KafkaHandler
31      public void handle(ProductCreatedEvent productCreatedEvent) {
32          LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
33
34          String requestUrl = "http://localhost:8082/response/200";
35
36          try {
37              ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, Strin
38
39              if (response.getStatusCode().value() == HttpStatus.OK.value()) {
40                  LOGGER.info("Received response from a remote service: " + response.getBody());
41              }
42          } catch (ResourceAccessException ex) {
43              LOGGER.error(ex.getMessage());
44              throw new RetryableException(ex);
45          } catch(HttpServerErrorException ex) {
46              LOGGER.error(ex.getMessage());
47              throw new NotRetryableException(ex);
48          } catch(Exception ex) {
49              LOGGER.error(ex.getMessage());
50              throw new NotRetryableException(ex);
51          }
52
53      }
54
55  }
56
```

Console × Problems Debug Shell Boot Dashboard

```
2023-12-18T09:37:48.484-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
2023-12-18T09:37:48.484-05:00  INFO 3851 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : Kafka
2023-12-18T09:37:48.489-05:00  INFO 3851 --- [ad | producer-1] org.apache.kafka.clients.Metadata         : [Produ
2023-12-18T09:37:48.489-05:00  INFO 3851 --- [ad | producer-1] o.a.k.c.p.internals.TransactionManager   : [Produ
2023-12-18T09:38:44.769-05:00  INFO 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Receiv
2023-12-18T09:38:44.772-05:00 ERROR 3851 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : I/O er
```

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

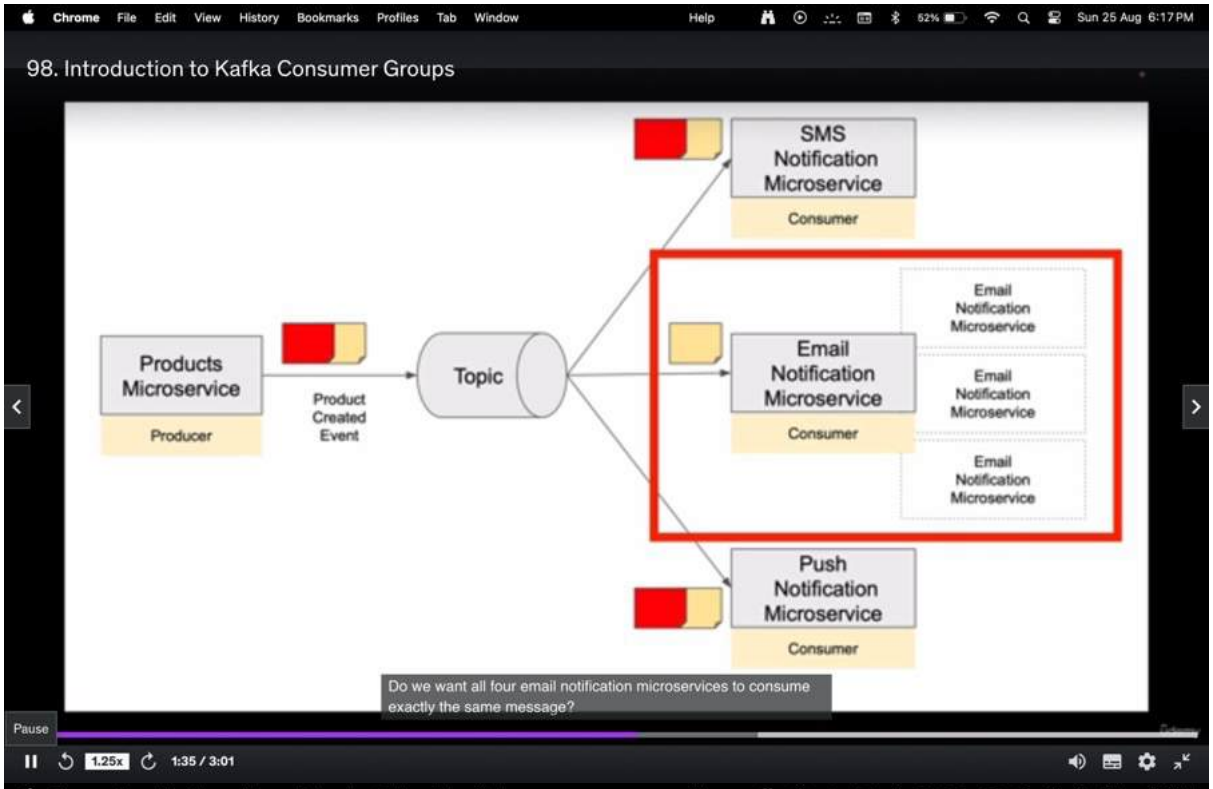
ProductCreatedEventHandler.java × KafkaConsumerConfiguration.java

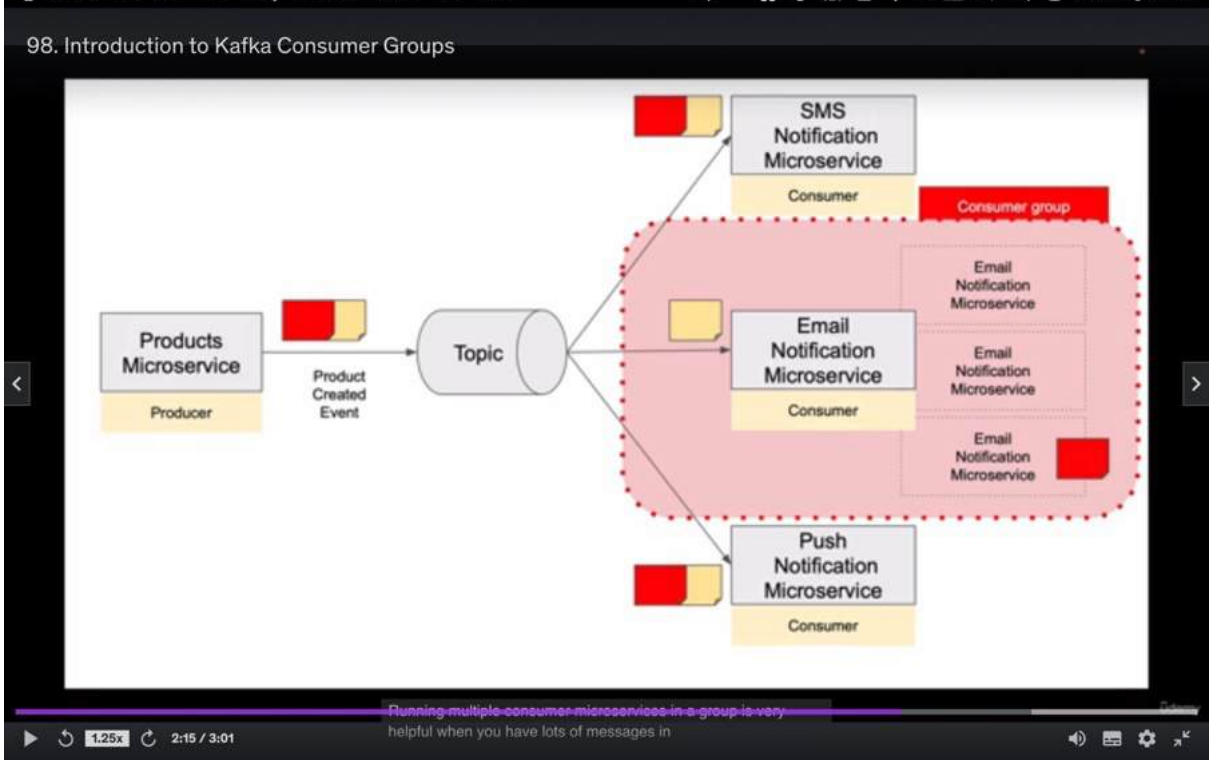
```java
29
30●     @KafkaHandler
31      public void handle(ProductCreatedEvent productCreatedEvent) {
32          LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());
33
34          String requestUrl = "http://localhost:8082/response/200";
35
36          try {
37              ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, Strin
38
39              if (response.getStatusCode().value() == HttpStatus.OK.value()) {
40                  LOGGER.info("Received response from a remote service: " + response.getBody());
41              }
43              LOGGER.error(ex.getMessage());
44              throw new RetryableException(ex);
45          } catch(HttpServerErrorException ex) {
46              LOGGER.error(ex.getMessage());
47              throw new NotRetryableException(ex);
48          } catch(Exception ex) {
49              LOGGER.error(ex.getMessage());
50              throw new NotRetryableException(ex);
51          }
52
53      }
54
55  }
56
```

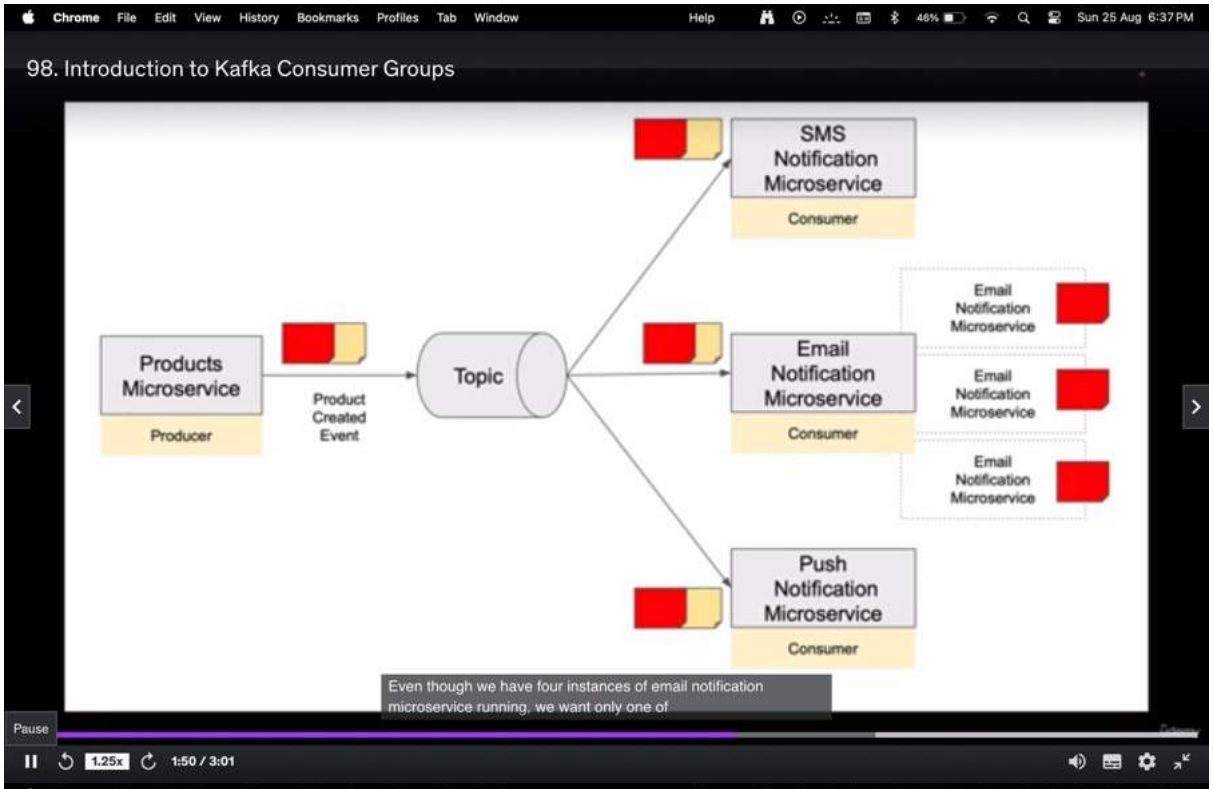
Console × Problems Debug Shell Boot Dashboard

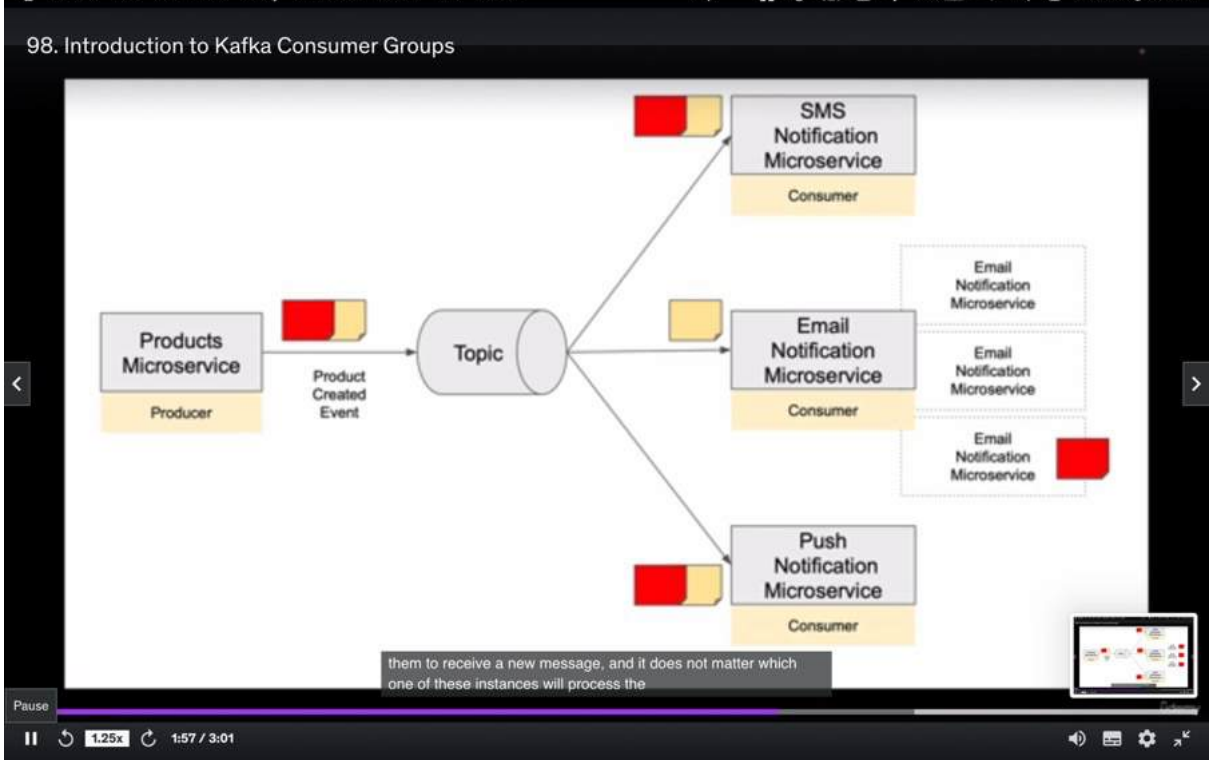
```
2023-12-18T09:39:04.026-05:00  INFO 4310 --- [           main] w.s.c.ServletWebServerApplicationContext : Root W
2023-12-18T09:39:04.257-05:00  INFO 4310 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat
2023-12-18T09:39:04.264-05:00  INFO 4310 --- [           main] c.a.w.m.MockserviceApplication
2023-12-18T09:39:17.478-05:00  INFO 4310 --- [nio-8082-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
2023-12-18T09:39:17.478-05:00  INFO 4310 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServlet
2023-12-18T09:39:17.480-05:00  INFO 4310 --- [nio-8082-exec-1] o.s.web.servlet.DispatcherServlet
```

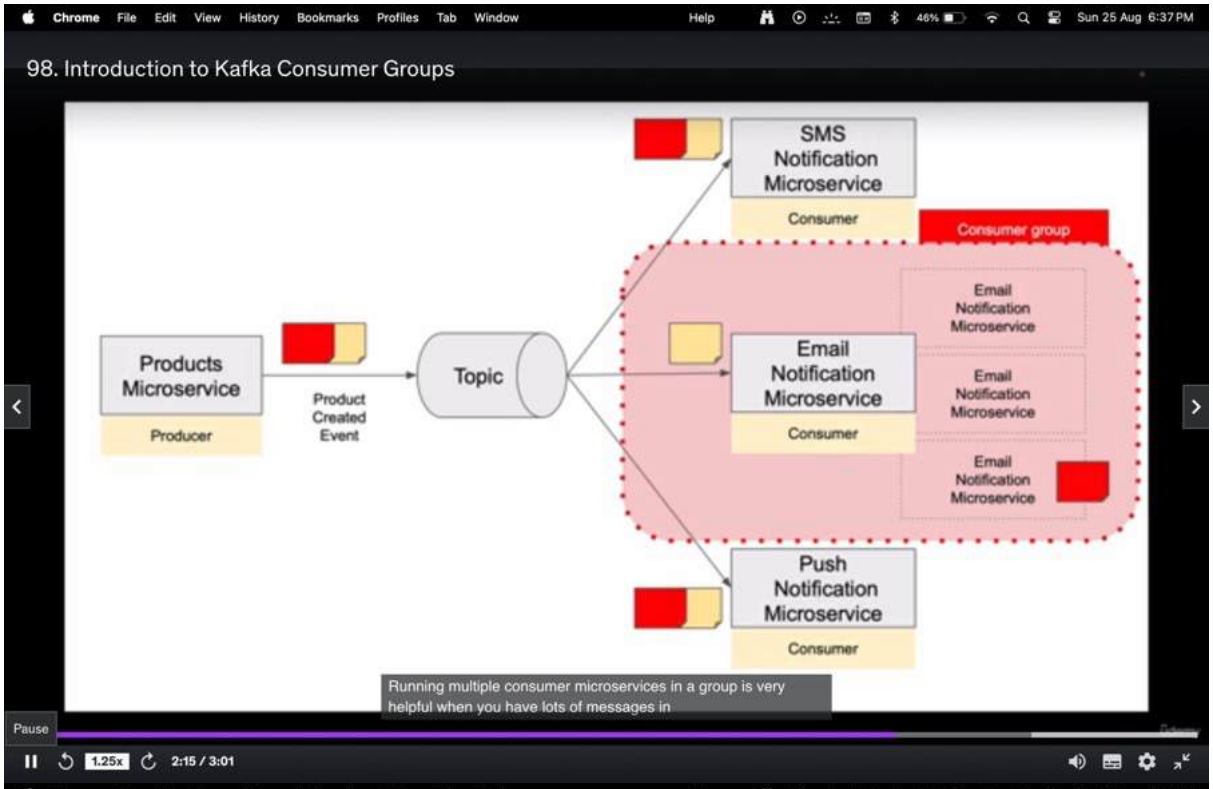
The next time this message is consumed, we have a successful response.

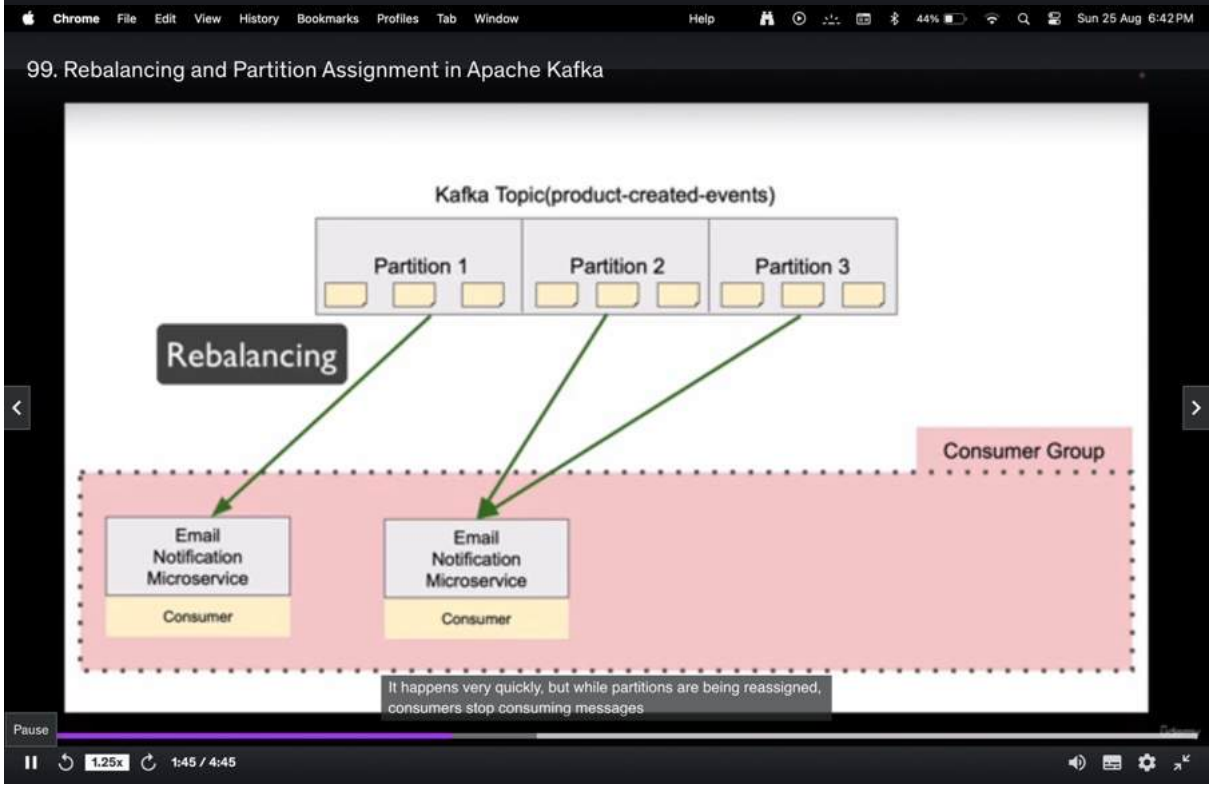

Chrome File Edit View History Bookmarks Profiles Tab Window Help 52% Sun 25 Aug 6:17 PM
98. Introduction to Kafka Consumer Groups
SMS Notification Microservice
Consumer
Products Microservice
Producer
Product Created Event
Topic
Email Notification Microservice
Consumer
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Push Notification Microservice
Consumer
Do we want all four email notification microservices to consume exactly the same message?
Pause
1.25x
1:35 / 3:01

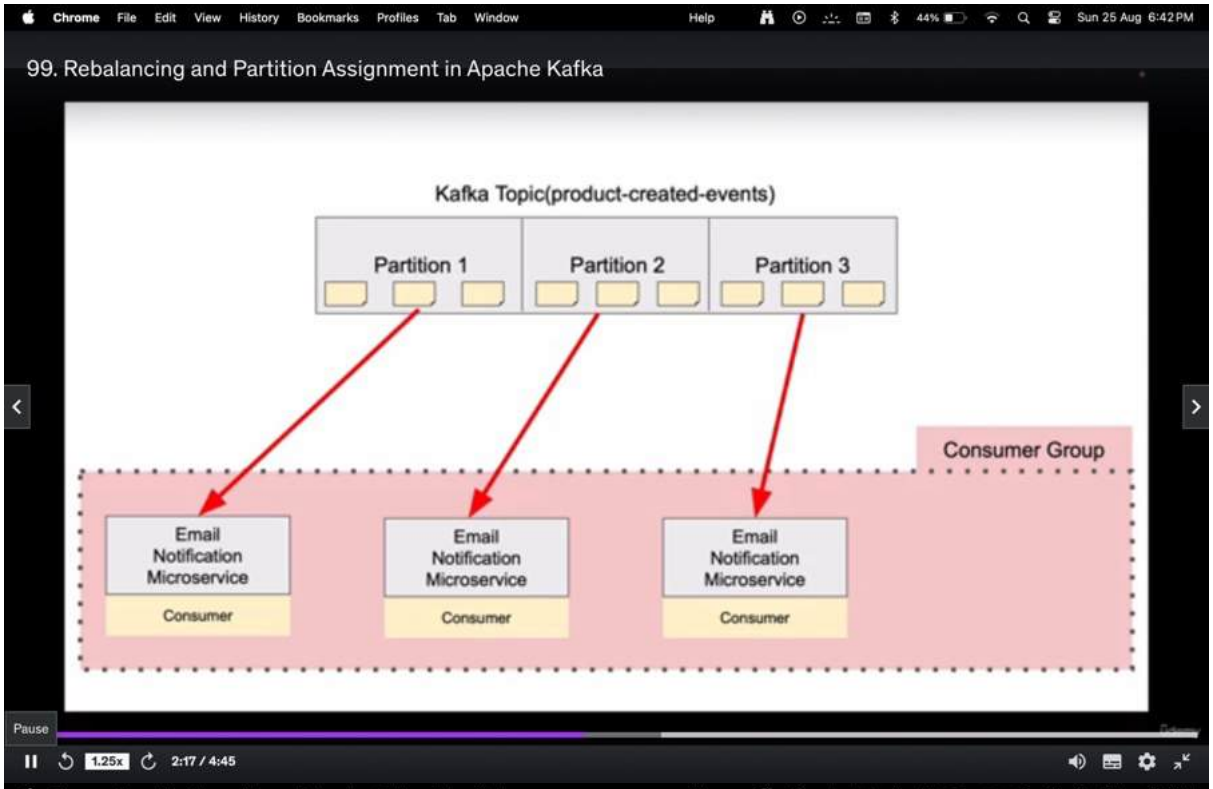

Chrome File Edit View History Bookmarks Profiles Tab Window Help 52% Sun 25 Aug 6:18 PM
98. Introduction to Kafka Consumer Groups
SMS Notification Microservice
Consumer
Consumer group
Products Microservice
Producer
Product Created Event
Topic
Email Notification Microservice
Consumer
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Push Notification Microservice
Consumer
Running multiple consumer microservices in a group is very helpful when you have lots of messages in
1.25x
2:15 / 3:01

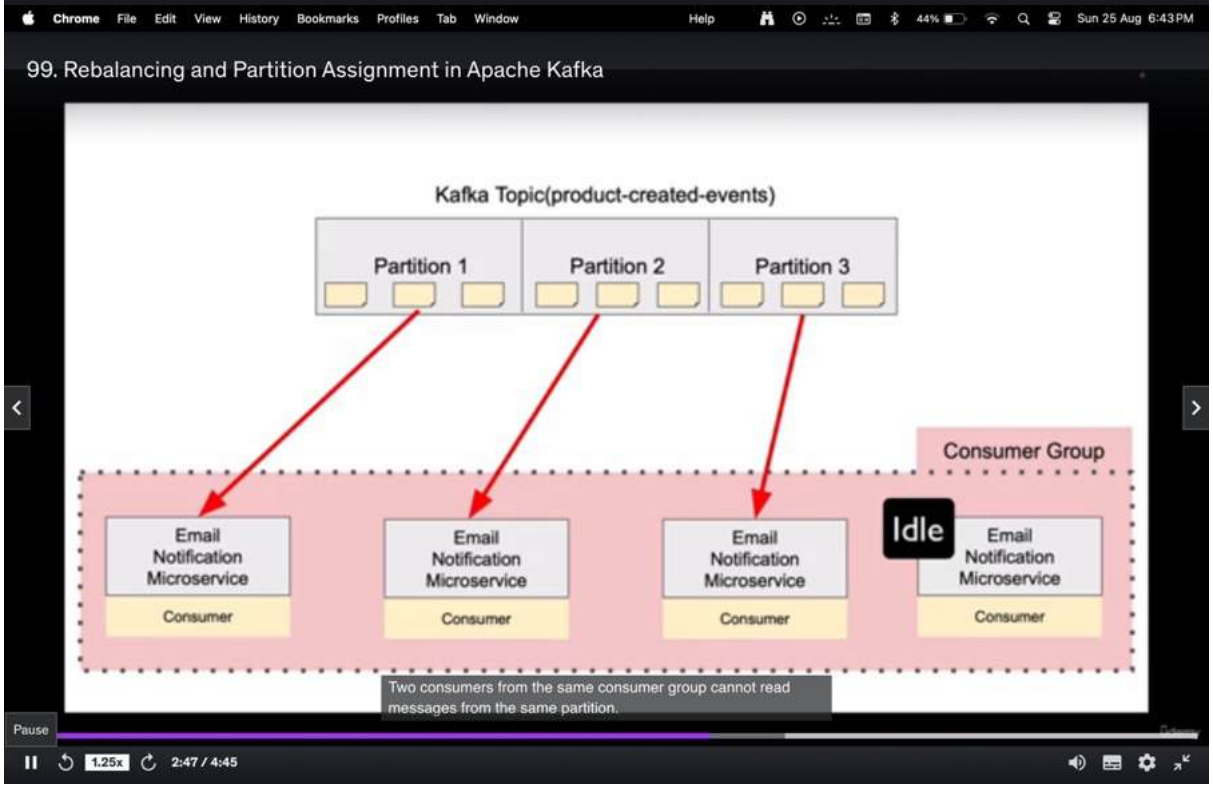

Chrome File Edit View History Bookmarks Profiles Tab Window Help 46% Sun 25 Aug 6:37 PM
98. Introduction to Kafka Consumer Groups
Products Microservice
Producer
Product Created Event
Topic
SMS Notification Microservice
Consumer
Email Notification Microservice
Consumer
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Push Notification Microservice
Consumer
Even though we have four instances of email notification microservice running, we want only one of
Pause
1.25x
1:50 / 3:01

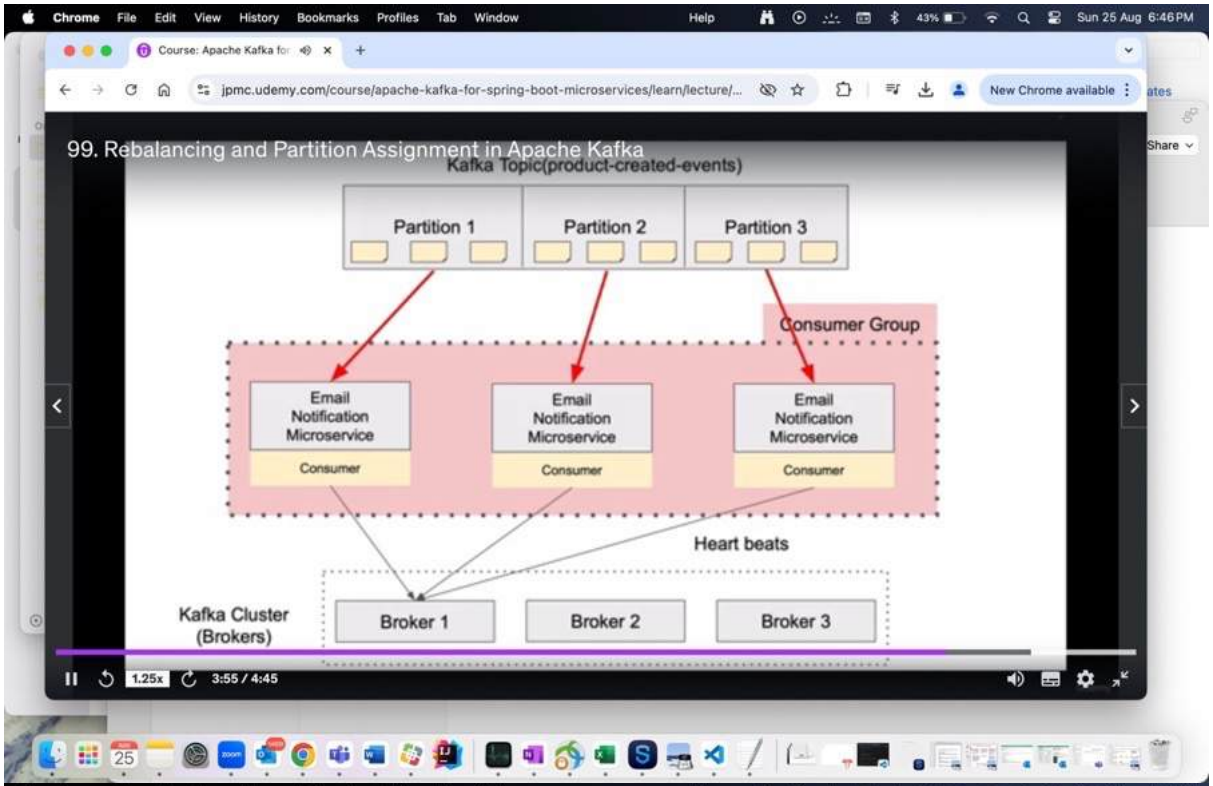

Chrome File Edit View History Bookmarks Profiles Tab Window Help 46% Sun 25 Aug 6:37 PM
98. Introduction to Kafka Consumer Groups
Products Microservice
Producer
Product Created Event
Topic
SMS Notification Microservice
Consumer
Email Notification Microservice
Consumer
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Push Notification Microservice
Consumer
them to receive a new message, and it does not matter which one of these instances will process the
Pause
1.25x
1:57 / 3:01

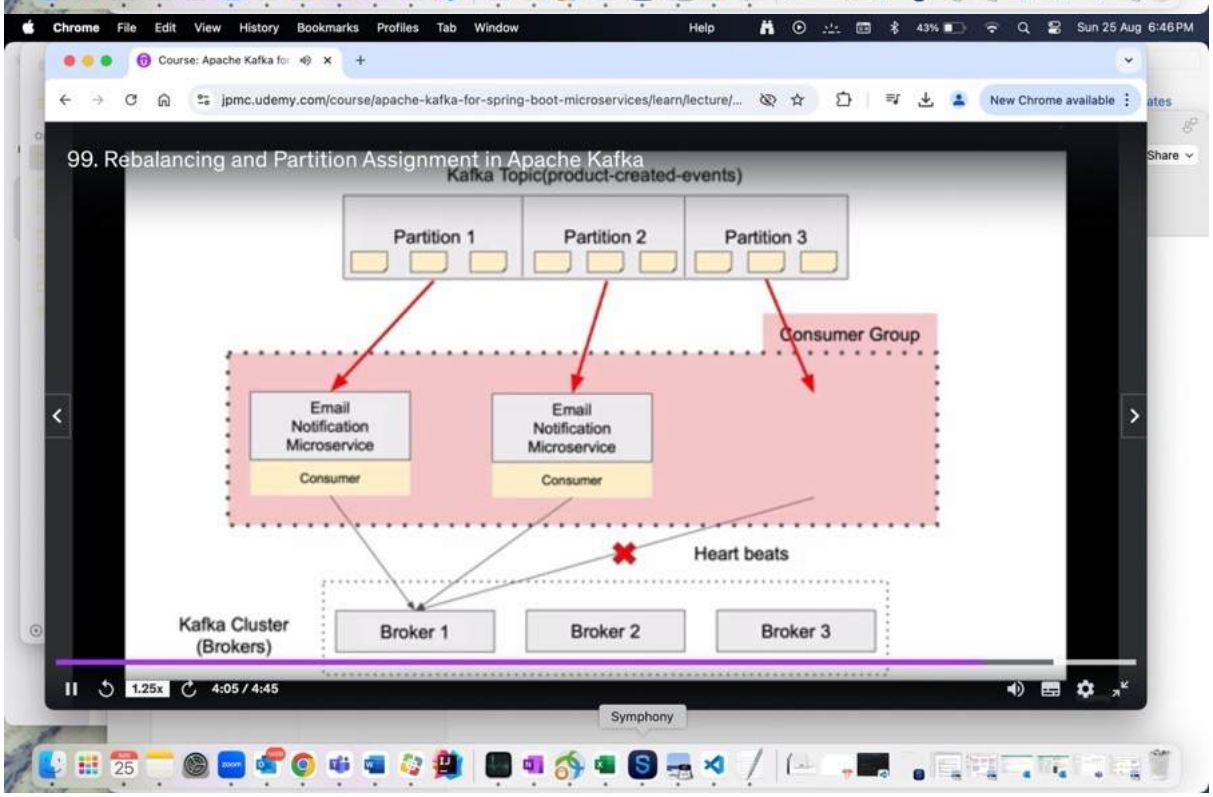

Chrome File Edit View History Bookmarks Profiles Tab Window Help 46% Sun 25 Aug 6:37 PM
98. Introduction to Kafka Consumer Groups
SMS Notification Microservice
Consumer
Consumer group
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Email Notification Microservice
Consumer
Products Microservice
Producer
Product Created Event
Topic
Push Notification Microservice
Consumer
Running multiple consumer microservices in a group is very helpful when you have lots of messages in
Pause
1.25x
2:15 / 3:01


Chrome File Edit View History Bookmarks Profiles Tab Window Help 44% Sun 25 Aug 6:42 PM
99. Rebalancing and Partition Assignment in Apache Kafka
Kafka Topic(product-created-events)
Partition 1
Partition 2
Partition 3
Rebalancing
Consumer Group
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
It happens very quickly, but while partitions are being reassigned, consumers stop consuming messages
Pause
1.25x
1:45 / 4:45


Chrome File Edit View History Bookmarks Profiles Tab Window Help 44% Sun 25 Aug 6:42 PM
99. Rebalancing and Partition Assignment in Apache Kafka
Kafka Topic(product-created-events)
Partition 1
Partition 2
Partition 3
Consumer Group
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Pause
1.25x
2:17 / 4:45


Chrome File Edit View History Bookmarks Profiles Tab Window Help 44% Sun 25 Aug 6:43 PM
99. Rebalancing and Partition Assignment in Apache Kafka
Kafka Topic(product-created-events)
Partition 1
Partition 2
Partition 3
Consumer Group
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Idle
Email Notification Microservice
Consumer
Two consumers from the same consumer group cannot read messages from the same partition.
Pause
1.25x
2:47 / 4:45


99. Rebalancing and Partition Assignment in Apache Kafka
Kafka Topic(product-created-events)
Partition 1
Partition 2
Partition 3
Consumer Group
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Heart beats
Kafka Cluster (Brokers)
Broker 1
Broker 2
Broker 3
1.25x
3:55 / 4:45


99. Rebalancing and Partition Assignment in Apache Kafka
Kafka Topic(product-created-events)
Partition 1
Partition 2
Partition 3
Consumer Group
Email Notification Microservice
Consumer
Email Notification Microservice
Consumer
Heart beats
Kafka Cluster (Brokers)
Broker 1
Broker 2
Broker 3
1.25x
4:05 / 4:45

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Sun 25 Aug 6:48 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/...

New Chrome available

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Tue Dec 19 12:52 PM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics = "product-created-events-topic", groupId="product-created-events")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch(HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch(Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }
```

Writable Smart Insert 20 : 88 : 912

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Sun 25 Aug 6:48 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/...

New Chrome available

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Tue Dec 19 12:53 PM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

ProductCreatedEventHandler.java

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch(HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch(Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }
```

Writable Smart Insert 20 : 98 : 880

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/...

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsumerConfiguration.java - Spring Tool Suite 4

ProductCreatedEventHandler.java | KafkaConsumerConfiguration.java

```java
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();
        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate),
                new FixedBackOff(5000,3));
        errorHandler.addNotRetryableExceptions(NotRetryableException.class);
        errorHandler.addRetryableExceptions(RetryableException.class);

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);
        factory.setCommonErrorHandler(errorHandler);

        return factory;
    }

    @Bean
```

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/...

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/resources/application.properties - Spring Tool Suite 4

ProductCreatedEventHandler.java | KafkaConsumerConfiguration.java | application.properties

```properties
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
spring.kafka.consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.ws.core
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help 29% Sun 25 Aug 8:19 PM

SpringToolSuite4 File Edit Source Navigate Search Project Run Window Help

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/resources/application.properties

CRITICAL
Network Drive Error
An error occurred while connecting to "Home Drive".

ProductCreatedEventHandler.java | KafkaConsumerConfiguration.java | *application.properties

```
1 server.port=0
2 spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
3 #spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
4 #spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
5 consumer.group-id=product-created-events
6 spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.ws.core
```

Writable Insert 5 : 1 : 285

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 29% Sun 25 Aug 8:19 PM

100. Assigning Microservice to a consumer group in Kafka

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/KafkaConsu

CRITICAL
Network Drive Error
An error occurred while connecting to "Home Drive".

ProductCreatedEventHandler.java | *KafkaConsumerConfiguration.java | *application.properties

```
31 public class KafkaConsumerConfiguration {
32
33     @Autowired
34     Environment environment;
35
36     @Bean
37     ConsumerFactory<String, Object> consumerFactory() {
38         Map<String, Object> config = new HashMap<>();
39         config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
40                 environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
41         config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
42         config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
43         config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
44         config.put(JsonDeserializer.TRUSTED_PACKAGES,
45                 environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));
46         config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("consumer.group-id"));
47
48         return new DefaultKafkaConsumerFactory<>(config);
49     }
50
51     @Bean
52     ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
53             ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {
54
55         DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate),
56                 new FixedBackOff(5000,3));
57         errorHandler.addNotRetryableExceptions(NotRetryableException.class);
58         errorHandler.addRetryableExceptions(RetryableException.class);
59
60         ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
61         factory.setConsumerFactory(consumerFactory);
62         factory.setCommonErrorHandler(errorHandler);
63
64         return factory;
65     }
66
67     @Bean
```

1.25x 4:01 / 4:23

101. Starting up more Microservices

Network Drive Error
An error occurred while connecting to "Home Drive".

ProductCreatedEventHandler.java | KafkaConsumerConfiguration.java | application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.ws.core
```

1.25x 1:23 / 4:26

```
Last login: Mon Dec 18 09:18:01 on ttys003
sergeykargopolov@Sergeys-MacBook-Pro kafka % cd "/Users/sergeykargopolov/video-courses/Event-Driven Microservices w
ith Apache Kafka/EmailNotificationMicroservice"
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % ls
HELP.md         mvnw            mvnw.cmd        pom.xml         src             target
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
```

101. Starting up more Microservices

```
2023-12-19T14:24:11.662-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 4: {consumer-product-created-events-1-86cfa81d-b337-4e0d-8b12-5a70d6cba450=Assignment(partitions=[product-cr
eated-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])}
2023-12-19T14:24:11.723-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=4, memberId='consumer-product-created-events-1-86cfa81d-b337-4e0d-8b12-5a70d6cba450', pr
otocol='range'}
2023-12-19T14:24:11.723-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])
2023-12-19T14:24:11.725-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2
2023-12-19T14:24:11.750-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:24:11.751-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T14:24:11.751-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:24:11.751-05:00  INFO 24568 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1, product-creat
ed-events-topic-2]
```

1.25x 2:47 / 4:26

```
Last login: Tue Dec 19 14:24:37 on ttys007
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % ls
HELP.md         mvnw            mvnw.cmd        pom.xml         src             target
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -------< com.appsdeveloperblog.ws:EmailNotificationMicroservice >-------
[INFO] Building EmailNotificationMicroservice 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot-maven-plugin:3.2.0:run (default-cli) > test-compile @ EmailNotificationMicroservice >>>
```

```
2023-12-19T14:34:42.040-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 10: {consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])}
2023-12-19T14:34:42.043-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=10, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:34:42.044-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])
2023-12-19T14:34:42.045-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2
2023-12-19T14:34:42.052-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le

2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1, product-creat
ed-events-topic-2]
```

```
e to: consumer pro-actively leaving the group
2023-12-19T14:34:07.486-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:07.486-05:00  INFO 24628 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:34:07.487-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:34:07.487-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:34:08.159-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:34:08.160-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  08:06 min
[INFO] Finished at: 2023-12-19T14:34:08-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
```

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:36:08.862-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:36:09.063-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=11, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac',
 protocol='range'}
2023-12-19T14:36:09.067-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=11, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac', p
rotocol='range'}
2023-12-19T14:36:09.068-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1])
2023-12-19T14:36:09.070-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1
2023-12-19T14:36:09.080-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:36:09.080-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
2023-12-19T14:36:09.081-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1]
```

```
created-events: partitions revoked: [product-created-events-topic-0, product-created-events-topic-1, product-create
d-events-topic-2]
2023-12-19T14:36:09.061-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:36:09.063-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=11, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0',
 protocol='range'}
2023-12-19T14:36:09.063-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 11: {consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1]), consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-
d34e9134cef0=Assignment(partitions=[product-created-events-topic-2])}
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=11, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:36:09.068-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
2023-12-19T14:36:09.068-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-2]
```

```
e to: consumer pro-actively leaving the group
2023-12-19T14:34:03.560-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:03.560-05:00  INFO 24661 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:34:03.561-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:34:03.561-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:34:04.167-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:34:04.168-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  07:13 min
[INFO] Finished at: 2023-12-19T14:34:04-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
```

```
2023-12-19T14:37:23.634-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: need to
 re-join with the given member-id: consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab',
 protocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab', p
rotocol='range'}
2023-12-19T14:37:24.091-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:37:24.093-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:37:24.100-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLead
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:37:24.101-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : pro
created-events: partitions assigned: [product-created-events-topic-2]
```

```
2023-12-19T14:37:23.634-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: need to
 re-join with the given member-id: consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab',
 protocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab', p
rotocol='range'}
2023-12-19T14:37:24.091-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:37:24.093-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:37:24.100-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:37:24.101-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-2]
```

```
2023-12-19T14:34:42.040-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 10: {consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])}
2023-12-19T14:34:42.043-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=10, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:34:42.044-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])
2023-12-19T14:34:42.045-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2
2023-12-19T14:34:42.052-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le

2023-12-19T14:34:42.053-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1, product-creat
ed-events-topic-2]
```

```
e to: consumer pro-actively leaving the group
2023-12-19T14:34:07.486-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:07.486-05:00  INFO 24628 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:34:07.487-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:34:07.487-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:34:08.156-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:34:08.159-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:34:08.160-05:00  INFO 24628 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  08:06 min
[INFO] Finished at: 2023-12-19T14:34:08-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
```

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:36:08.862-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:36:09.063-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=11, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac',
 protocol='range'}
2023-12-19T14:36:09.067-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=11, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac', p
rotocol='range'}
2023-12-19T14:36:09.068-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1])
2023-12-19T14:36:09.070-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1
2023-12-19T14:36:09.080-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:36:09.080-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
2023-12-19T14:36:09.081-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1]
```

```
created-events: partitions revoked: [product-created-events-topic-0, product-created-events-topic-1, product-create
d-events-topic-2]
2023-12-19T14:36:09.061-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:36:09.063-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=11, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0',
 protocol='range'}
2023-12-19T14:36:09.063-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 11: {consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1]), consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-
d34e9134cef0=Assignment(partitions=[product-created-events-topic-2])}
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=11, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:36:09.066-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:36:09.068-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
2023-12-19T14:36:09.068-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-2]
```

```
e to: consumer pro-actively leaving the group
2023-12-19T14:34:03.560-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:03.560-05:00  INFO 24661 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:34:03.561-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:34:03.561-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:34:04.163-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:34:04.167-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:34:04.168-05:00  INFO 24661 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  07:13 min
[INFO] Finished at: 2023-12-19T14:34:04-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
```

```
2023-12-19T14:37:23.634-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: need to
 re-join with the given member-id: consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab',
 protocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab', p
rotocol='range'}
2023-12-19T14:37:24.091-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:37:24.093-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:37:24.100-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLead
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:37:24.101-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : pro
created-events: partitions assigned: [product-created-events-topic-2]
```

```
2023-12-19T14:37:23.634-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: need to
 re-join with the given member-id: consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T14:37:23.635-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab',
 protocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-ce5f433e-3f4f-447e-8ffc-8d3cba81ceab', p
rotocol='range'}
2023-12-19T14:37:24.091-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T14:37:24.093-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T14:37:24.100-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:37:24.101-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-2]
```

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: group i
s already rebalancing
2023-12-19T14:37:24.082-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Revoke previously assigned partitions
 product-created-events-topic-0, product-created-events-topic-1
2023-12-19T14:37:24.083-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions revoked: [product-created-events-topic-0, product-created-events-topic-1]
2023-12-19T14:37:24.083-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac',
 protocol='range'}
2023-12-19T14:37:24.088-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac', p
rotocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0])
2023-12-19T14:37:24.089-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0
2023-12-19T14:37:24.092-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLead
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:37:24.092-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : pro
created-events: partitions assigned: [product-created-events-topic-0]
```

```
created-events: partitions revoked: [product-created-events-topic-2]
2023-12-19T14:37:24.078-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T14:37:24.085-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=12, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0',
 protocol='range'}
2023-12-19T14:37:24.085-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 12: {consumer-product-created-events-1-0bb6dfd3-baa9-4d92-900f-22f1ba0d0bac=Assignment(partitions=[product-c
reated-events-topic-0]), consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0=Assignment(partitio
ns=[product-created-events-topic-1]), consumer-product-created-events-1-ce5f4233e-3f4f-447e-8ffc-8d3cba81ceab=Assign
ment(partitions=[product-created-events-topic-2])}
2023-12-19T14:37:24.088-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=12, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:37:24.089-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-1])
2023-12-19T14:37:24.089-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-1
2023-12-19T14:37:24.090-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T14:37:24.091-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-1]
```

102. Trying how partitions assignment works in Kafka

```
e to: consumer pro-actively leaving the group
2023-12-19T14:40:33.618-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:40:33.619-05:00  INFO 24811 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:40:33.620-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:40:33.620-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:40:34.065-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:40:34.065-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:40:34.065-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:40:34.070-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:40:34.071-05:00  INFO 24811 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  03:13 min
[INFO] Finished at: 2023-12-19T14:40:34-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice %
```

1.25x 4:31 / 5:00

102. Trying how partitions assignment works in Kafka

```
e to: consumer pro-actively leaving the group
2023-12-19T14:40:46.889-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:40:46.889-05:00  INFO 24789 --- [ntainer#0-0-C-1] fkaConsumerFactory$ExtendedKafkaConsumer : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Unsubscribed all topics or patterns a
nd assigned partitions
2023-12-19T14:40:46.890-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Resetting generation and member id du
e to: consumer pro-actively leaving the group
2023-12-19T14:40:46.890-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: consume
r pro-actively leaving the group
2023-12-19T14:40:47.381-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
scheduler closed
2023-12-19T14:40:47.381-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Closing
reporter org.apache.kafka.common.metrics.JmxReporter
2023-12-19T14:40:47.381-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.apache.kafka.common.metrics.Metrics    : Metrics
reporters closed
2023-12-19T14:40:47.386-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.a.kafka.common.utils.AppInfoParser     : App info
 kafka.consumer for consumer-product-created-events-1 unregistered
2023-12-19T14:40:47.386-05:00  INFO 24789 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: Consumer stopped
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  04:41 min
[INFO] Finished at: 2023-12-19T14:40:47-05:00
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice %
```

1.25x 4:37 / 5:00

102. Trying how partitions assignment works in Kafka

```
2023-12-19T14:40:48.134-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 14: {consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])}
2023-12-19T14:40:48.136-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=14, memberId='consumer-product-created-events-1-c0015a3e-190a-46e2-8afb-d34e9134cef0', p
rotocol='range'}
2023-12-19T14:40:48.136-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])
2023-12-19T14:40:48.136-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2
2023-12-19T14:40:48.137-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:40:48.137-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T14:40:48.138-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T14:40:48.138-05:00  INFO 24766 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1, product-creat
ed-events-topic-2]
```

1.25x 4:46 / 5:00

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

ProductCreatedEventHandler.java × KafkaConsumerConfiguration.java application.properties

```java
@Component
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(ProductCreatedEvent productCreatedEvent) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch (HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch (Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

    }
```

103. Multiple consumers consuming messages from Kafka topic

```
2023-12-19T19:48:05.672-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 16: {consumer-product-created-events-1-b377e52f-9c46-4f04-9c2e-81ec9ddb4121=Assignment(partitions=[product-c
reated-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])}
2023-12-19T19:48:05.682-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=16, memberId='consumer-product-created-events-1-b377e52f-9c46-4f04-9c2e-81ec9ddb4121', p
rotocol='range'}
2023-12-19T19:48:05.683-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2])
2023-12-19T19:48:05.684-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0, product-created-events-topic-1, product-created-events-topic-2
2023-12-19T19:48:05.695-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:48:05.695-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T19:48:05.695-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:48:05.696-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0, product-created-events-topic-1, product-creat
ed-events-topic-2]
```

1.25x 1:30 / 4:22

103. Multiple consumers consuming messages from Kafka topic

```
[INFO] ------------------------------------------------------------------------
sergeykargopolov@Sergeys-MacBook-Pro EmailNotificationMicroservice % mvn spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -------< com.appsdeveloperblog.ws:EmailNotificationMicroservice >-------
[INFO] Building EmailNotificationMicroservice 0.0.1-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] >>> spring-boot-maven-plugin:3.2.0:run (default-cli) > test-compile @ EmailNotificationMicroservice >>>
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:resources (default-resources) @ EmailNotificationMicroservice ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 0 resource from src/main/resources to target/classes
[INFO]
[INFO] --- maven-compiler-plugin:3.11.0:compile (default-compile) @ EmailNotificationMicroservice ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:3.3.1:testResources (default-testResources) @ EmailNotificationMicroservice ---
[INFO] skip non existing resourceDirectory /Users/sergeykargopolov/video-courses/Event-Driven Microservices with Ap
ache Kafka/EmailNotificationMicroservice/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.11.0:testCompile (default-testCompile) @ EmailNotificationMicroservice ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< spring-boot-maven-plugin:3.2.0:run (default-cli) < test-compile @ EmailNotificationMicroservice <<<
[INFO]
[INFO]
[INFO] --- spring-boot-maven-plugin:3.2.0:run (default-cli) @ EmailNotificationMicroservice ---
[INFO] Attaching agents: 
```

1.25x 1:52 / 4:22

103. Multiple consumers consuming messages from Kafka topic

POST http://localhost:53373/products

```
{
    "title":"iPad Pro",
    "price":1250,
    "quantity":19
}
```

Status: 201 Created Time: 802 ms Size: 205 B

```
08d3a958-361d-4be7-a4d5-5c7be2a80083
```

1.25x 3:21 / 4:22

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T19:49:03.036-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T19:49:05.696-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8',
 protocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8', p
rotocol='range'}
2023-12-19T19:49:05.704-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0])
2023-12-19T19:49:05.705-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0
2023-12-19T19:49:05.712-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:49:05.713-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0]
2023-12-19T19:51:36.011-05:00  INFO 28143 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received
 a new event: iPad Pro with productId: 08d3a958-361d-4be7-a4d5-5c7be2a80083
2023-12-19T19:51:36.380-05:00  INFO 28143 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Received
 response from a remote service: 200
```

103. Multiple consumers consuming messages from Kafka topic


```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T19:49:03.036-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T19:49:05.696-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8',
 protocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8', p
rotocol='range'}
2023-12-19T19:49:05.704-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0])
2023-12-19T19:49:05.705-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0
2023-12-19T19:49:05.712-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:49:05.713-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-0]
2023-12-19T19:51:36.011-05:00  INFO 28143 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEvent
 a new event: iPad Pro with productId: 08d3a958-361d-4be7-a4d5-5c7be2a80083
2023-12-19T19:51:36.380-05:00  INFO 28143 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEvent
 response from a remote service: 200
```


1.25x 3:36 / 4:22

103. Multiple consumers consuming messages from Kafka topic

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: rebalan
ce failed due to 'The group member needs to have a valid member id before actually entering a consumer group.' (Mem
berIdRequiredException)
2023-12-19T19:49:03.036-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T19:49:05.696-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8',
 protocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=18, memberId='consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8', p
rotocol='range'}
2023-12-19T19:49:05.704-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-0])
2023-12-19T19:49:05.705-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-0
2023-12-19T19:49:05.712-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-0 to the committed offset FetchPosition{offset=2, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:49:05.713-05:00  INFO 28143 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-

2023-12-19T19:51:36.011-05:00  INFO 28143 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEven
 a new event: iPad Pro with productId: 08d3a958-361d-4be7-a4d5-5c7be2a80083

 response from a remote service: 200
```

1.25x 3:35 / 4:22

103. Multiple consumers consuming messages from Kafka topic

```
r clientId=consumer-product-created-events-1, groupId=product-created-events] Request joining group due to: group i
s already rebalancing
2023-12-19T19:49:05.693-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Revoke previously assigned partitions
 product-created-events-topic-2
2023-12-19T19:49:05.694-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions revoked: [product-created-events-topic-2]
2023-12-19T19:49:05.694-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T19:49:05.696-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=18, memberId='consumer-product-created-events-1-d3403375-23ae-4b33-a78f-7c9833813182',
 protocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=18, memberId='consumer-product-created-events-1-d3403375-23ae-4b33-a78f-7c9833813182', p
rotocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-2])
2023-12-19T19:49:05.701-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-2
2023-12-19T19:49:05.703-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-2 to the committed offset FetchPosition{offset=0, offsetEpoch=Optional.empty, currentLeader=Le
aderAndEpoch{leader=Optional[localhost:9094 (id: 2 rack: null)], epoch=0}}
2023-12-19T19:49:05.703-05:00  INFO 28127 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : product-
created-events: partitions assigned: [product-created-events-topic-2]
```

1.25x 3:58 / 4:22

103. Multiple consumers consuming messages from Kafka topic

```
created-events: partitions revoked: [product-created-events-topic-0, product-created-events-topic-1]
2023-12-19T19:49:05.692-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] (Re-)joining group
2023-12-19T19:49:05.696-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully joined group with genera
tion Generation{generationId=18, memberId='consumer-product-created-events-1-b377e52f-9c46-4f04-9c2e-81ec9ddb4121',
 protocol='range'}
2023-12-19T19:49:05.696-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Finished assignment for group at gene
ration 18: {consumer-product-created-events-1-7325c9b3-8574-4422-b15b-4b0d8cd951c8=Assignment(partitions=[product-c
reated-events-topic-0]), consumer-product-created-events-1-d3403375-23ae-4b33-a78f-7c9833813182=Assignment(partitio
ns=[product-created-events-topic-2]), consumer-product-created-events-1-b377e52f-9c46-4f04-9c2e-81ec9ddb4121=Assign
ment(partitions=[product-created-events-topic-1])}
2023-12-19T19:49:05.701-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Successfully synced group in generati
on Generation{generationId=18, memberId='consumer-product-created-events-1-b377e52f-9c46-4f04-9c2e-81ec9ddb4121', p
rotocol='range'}
2023-12-19T19:49:05.701-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Notifying assignor about the new Assi
gnment(partitions=[product-created-events-topic-1])
2023-12-19T19:49:05.701-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Adding newly assigned partitions: pro
duct-created-events-topic-1
2023-12-19T19:49:05.703-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator  : [Consume
r clientId=consumer-product-created-events-1, groupId=product-created-events] Setting offset for partition product-
created-events-topic-1 to the committed offset FetchPosition{offset=1, offsetEpoch=Optional.empty, currentLead
aderAndEpoch{leader=Optional[localhost:9092 (id: 1 rack: null)], epoch=0}}
2023-12-19T19:49:05.703-05:00  INFO 28108 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer    : pr
created-events: partitions assigned: [product-created-events-topic-1]
```

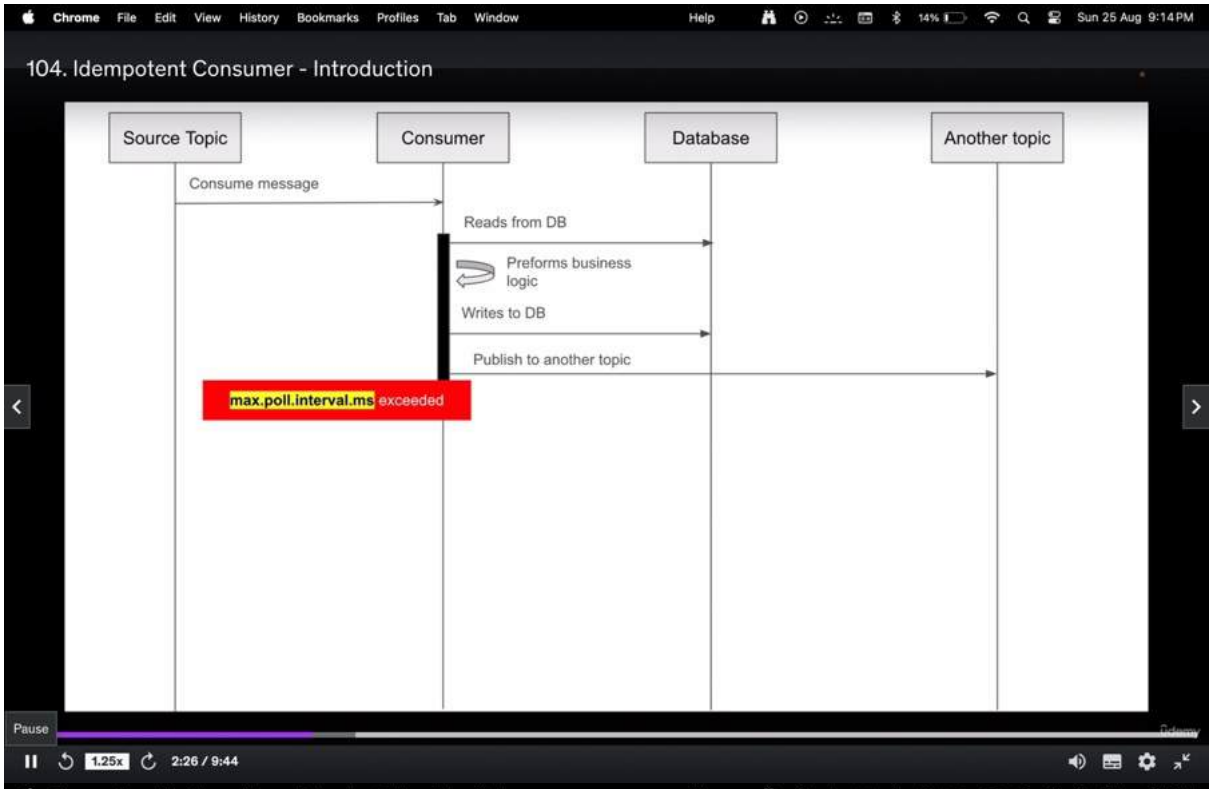
1.25x 4:04 / 4:22

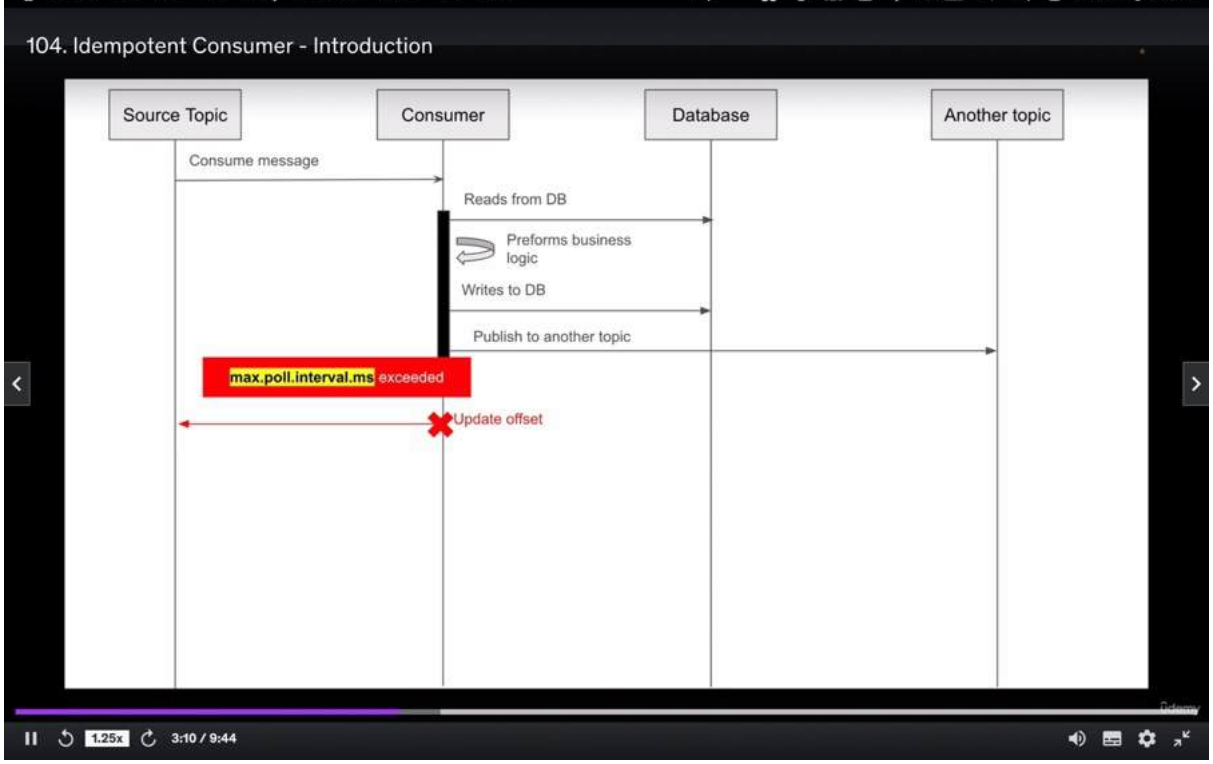
104. Idempotent Consumer - Introduction

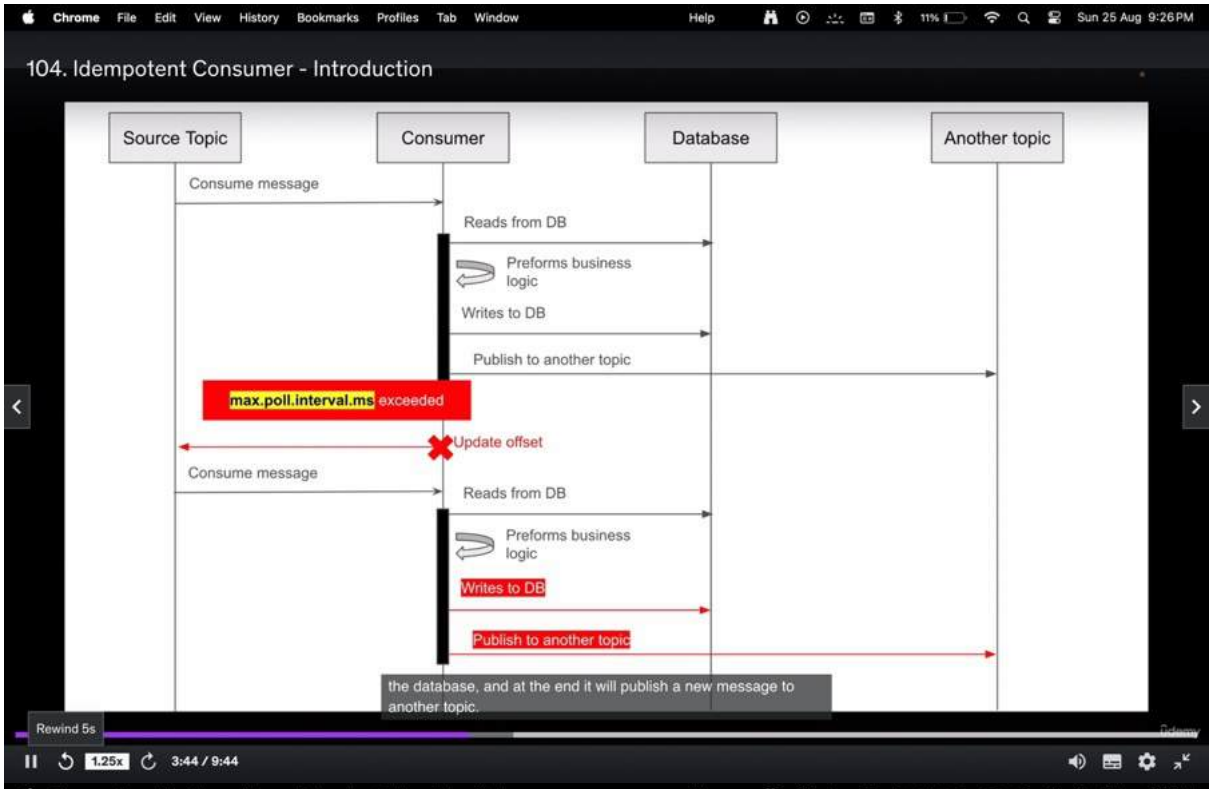
What is Idempotent Consumer?

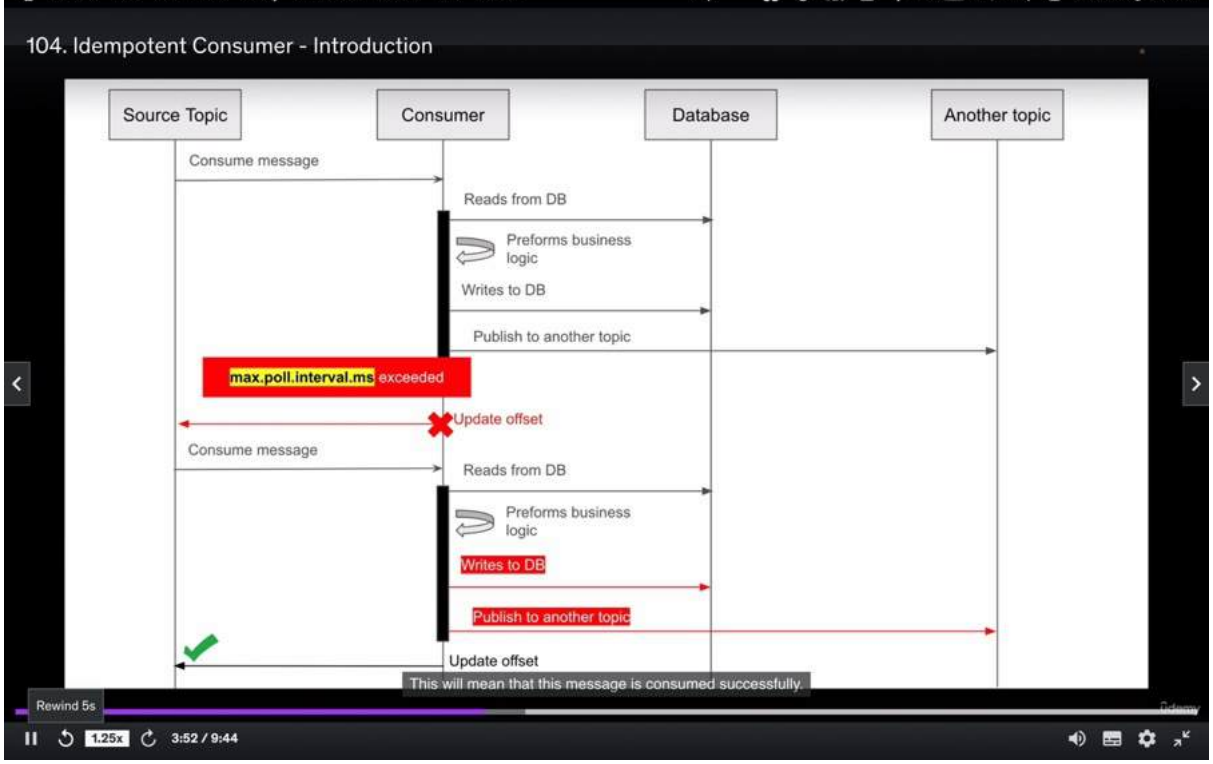
An idempotent Apache Kafka consumer is a consumer that can process the same message multiple times without causing any side effects or data inconsistencies.

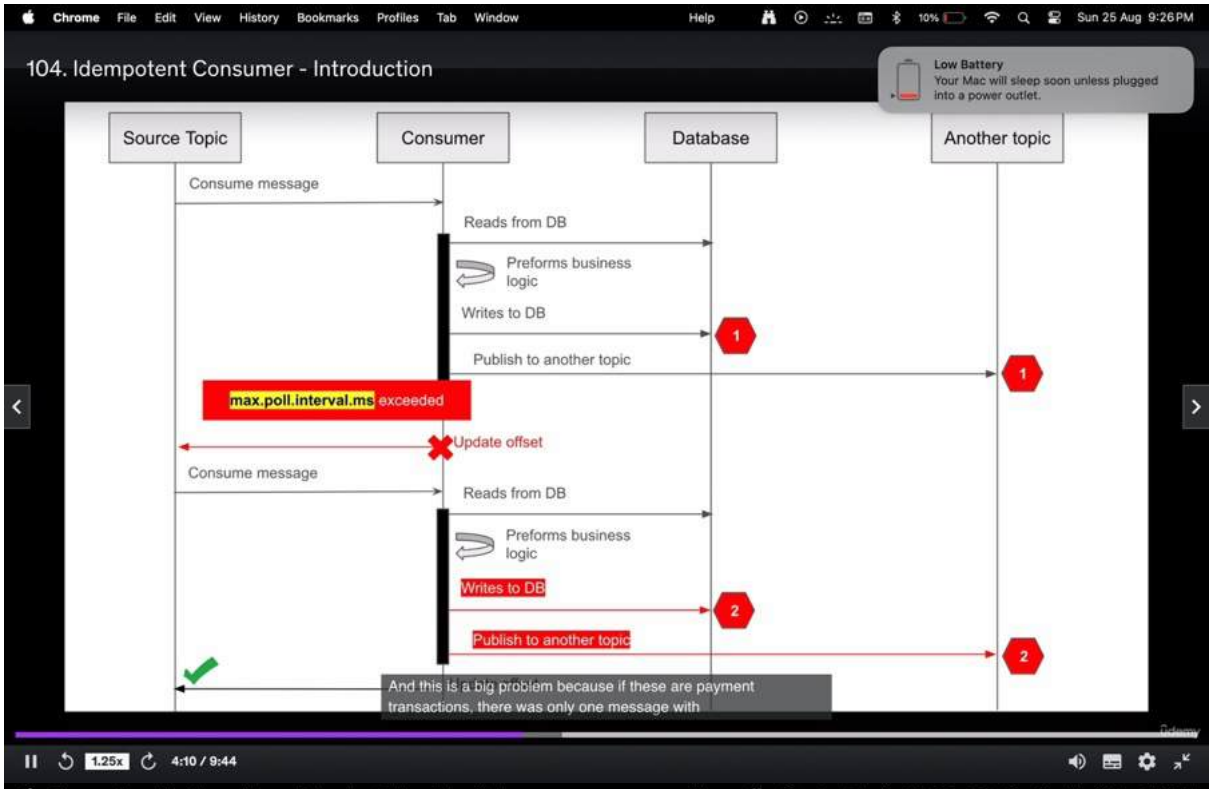
1.25x 0:38 / 9:44

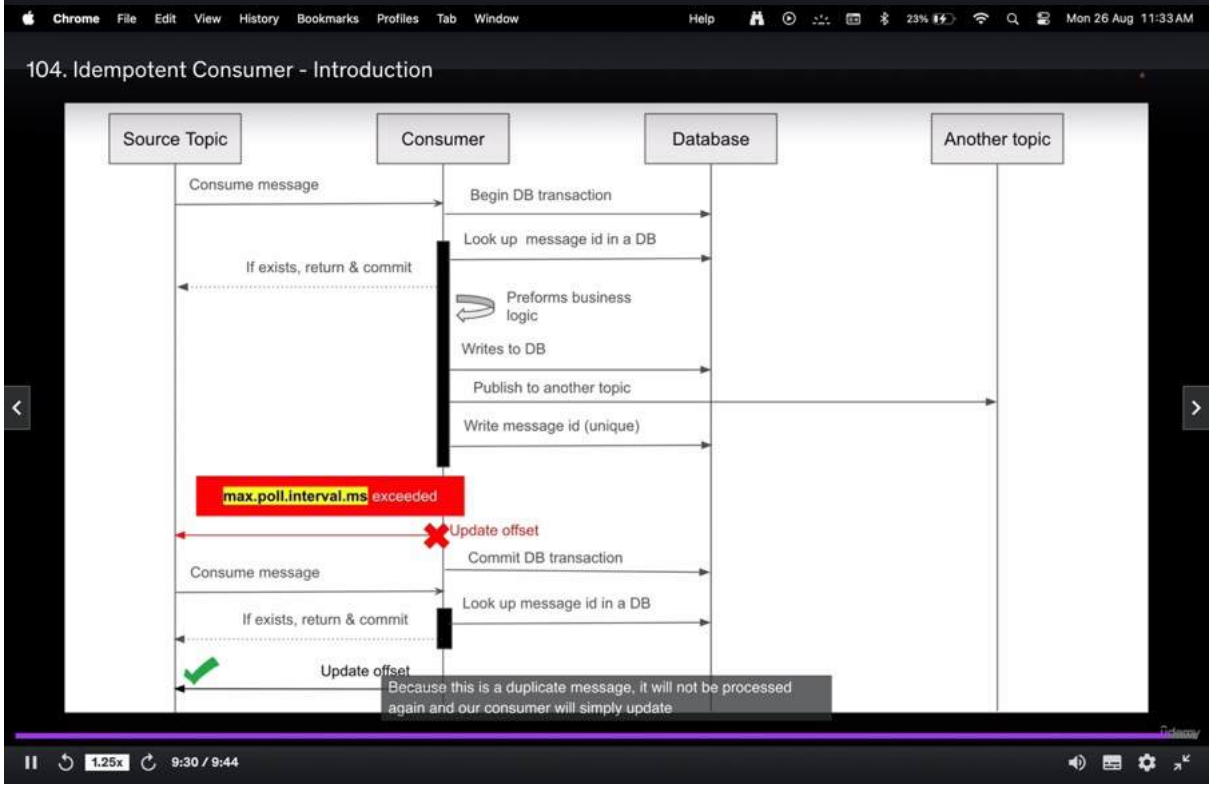

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 9:14 PM
104. Idempotent Consumer - Introduction
Source Topic
Consumer
Database
Another topic
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
max.poll.interval.ms exceeded
Pause
1.25x
2:26 / 9:44


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 9:21 PM
104. Idempotent Consumer - Introduction
Source Topic
Consumer
Database
Another topic
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
max.poll.interval.ms exceeded
Update offset
1.25x
3:10 / 9:44


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 9:26 PM
104. Idempotent Consumer - Introduction
Source Topic
Consumer
Database
Another topic
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
max.poll.interval.ms exceeded
Update offset
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
the database, and at the end it will publish a new message to another topic.
Rewind 5s
1.25x
3:44 / 9:44


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun 25 Aug 9:26 PM
104. Idempotent Consumer - Introduction
Source Topic
Consumer
Database
Another topic
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
max.poll.interval.ms exceeded
Update offset
Consume message
Reads from DB
Preforms business logic
Writes to DB
Publish to another topic
Update offset
This will mean that this message is consumed successfully.
Rewind 5s
1.25x
3:52 / 9:44


Chrome File Edit View History Bookmarks Profiles Tab Window Help 10% Sun 25 Aug 9:26 PM
104. Idempotent Consumer - Introduction
Low Battery
Your Mac will sleep soon unless plugged into a power outlet.
Source Topic
Consumer
Database
Another topic
Consume message
Reads from DB
Preforms business logic
Writes to DB
1
Publish to another topic
1
max.poll.interval.ms exceeded
Update offset
Consume message
Reads from DB
Preforms business logic
Writes to DB
2
Publish to another topic
2
And this is a big problem because if these are payment transactions, there was only one message with
1.25x 4:10 / 9:44


Chrome File Edit View History Bookmarks Profiles Tab Window Help 23% Mon 26 Aug 11:33 AM
104. Idempotent Consumer - Introduction
Source Topic
Consumer
Database
Another topic
Consume message
Begin DB transaction
Look up message id in a DB
If exists, return & commit
Preforms business logic
Writes to DB
Publish to another topic
Write message id (unique)
max.poll.interval.ms exceeded
Update offset
Commit DB transaction
Consume message
Look up message id in a DB
If exists, return & commit
Update offset
Because this is a duplicate message, it will not be processed again and our consumer will simply update
1.25x 9:30 / 9:44

105. Include a unique id into message header

```
47  * {@link RecordMetadata}
48  */
49 public class ProducerRecord<K, V> {
50
51     private final String topic;
52     private final Integer partition;
53     private final Headers headers;
54     private final K key;
55     private final V value;
56     private final Long timestamp;
57
58     /**
59      * Creates a record with a specified timestamp to be sent to a specified topic and partit
60      *
61      * @param topic The topic the record will be appended to
62      * @param partition The partition to which the record should be sent
63      * @param timestamp The timestamp of the record, in milliseconds since epoch. If null, th
64      *                  the timestamp using System.currentTimeMillis().
65      * @param key The key that will be included in the record
66      * @param value The record contents
67      * @param headers the headers that will be included in the record
68      */
69     public ProducerRecord(String topic, Integer partition, Long timestamp, K key, V value, It
70         if (topic == null)
71             throw new IllegalArgumentException("Topic cannot be null.");
72         if (timestamp != null && timestamp < 0)
73             throw new IllegalArgumentException(
74                     String.format("Invalid timestamp: %d. Timestamp should always be non-nega
75         if (partition != null && partition < 0)
76             throw new IllegalArgumentException(
77                     String.format("Invalid partition: %d. Partition number should always be r
78         this.topic = topic;
79         this.partition = partition;
80         this.key = key;
81         this.value = value;
```

So what we can do is we can create a producer record object outside of send method.

1.25x 2:23 / 4:26

105. Include a unique id into message header

```
46  * In either of the cases above, the timestamp that has actually been used will be returned i
47  * {@link RecordMetadata}
48  */
49 public class ProducerRecord<K, V> {
50
51     private final String topic;
52     private final Integer partition;
53     private final Headers headers;
54     private final K key;
55     private final V value;
56     private final Long timestamp;
57
58     /**
59      * Creates a record with a specified timestamp to be sent to a specified topic and partit
60      *
61      * @param topic The topic the record will be appended to
62      * @param partition The partition to which the record should be sent
63      * @param timestamp The timestamp of the record, in milliseconds since epoch. If null, th
64      *                  the timestamp using System.currentTimeMillis().
65      * @param key The key that will be included in the record
66      * @param value The record contents
67      * @param headers the headers that will be included in the record
68      */
69     public ProducerRecord(String topic, Integer partition, Long timestamp, K key, V value, It
70         if (topic == null)
71             throw new IllegalArgumentException("Topic cannot be null.");
72         if (timestamp != null && timestamp < 0)
73             throw new IllegalArgumentException(
74                     String.format("Invalid timestamp: %d. Timestamp should always be non-nega
75         if (partition != null && partition < 0)
76             throw new IllegalArgumentException(
77                     String.format("Invalid partition: %d. Partition number should always be r
78         this.topic = topic;
79         this.partition = partition;
80         this.key = key;
81         this.value = value;
82         this.headers = new RecordHeaders(headers);
83
```

As a constructor argument here.

1.25x 2:09 / 4:26

105. Include a unique id into message header

```
47
48
49 {
50
51
52 n;
53
54
55
56
57
58
59 cified timestamp to be sent to a specified topic and partition
60
61 record will be appended to
62 ion to which the record should be sent
63 amp of the record, in milliseconds since epoch. If null, the producer will assign
64 amp using System.currentTimeMillis().
65 l be included in the record
66 tents
67 that will be included in the record
68
69 opic, Integer partition, Long timestamp, K key, V value, Iterable<Header> headers) {
70
71 entException("Topic cannot be null.");
72 imestamp < 0)
73 entException(
74 "Invalid timestamp: %d. Timestamp should always be non-negative or null.", timestamp));
75 artition < 0)
76 entException(
77 "Invalid partition: %d. Partition number should always be non-negative or null.", partition)
78
79 ;
80
```

So if we want to work with message headers, we will need to use producer record class.

1.25x 2:12 / 4:26

105. Include a unique id into message header

```
47 * {@link RecordMetadata}
48 */
49 public class ProducerRecord<K, V> {
50
51     private final String topic;
52     private final Integer partition;
53     private final Headers headers;
54     private final K key;
55     private final V value;
56     private final Long timestamp;
57
58     /**
59      * Creates a record with a specified timestamp to be sent to a specified topic and partit
60      *
61      * @param topic The topic the record will be appended to
62      * @param partition The partition to which the record should be sent
63      * @param timestamp The timestamp of the record, in milliseconds since epoch. If null, th
64      *                  the timestamp using System.currentTimeMillis().
65      * @param key The key that will be included in the record
66      * @param value The record contents
67      * @param headers the headers that will be included in the record
68      */
69     public ProducerRecord(String topic, Integer partition, Long timestamp, K key, V value, It
70         if (topic == null)
71             throw new IllegalArgumentException("Topic cannot be null.");
72         if (timestamp != null && timestamp < 0)
73             throw new IllegalArgumentException(
74                     String.format("Invalid timestamp: %d. Timestamp should always be non-nega
75         if (partition != null && partition < 0)
76             throw new IllegalArgumentException(
77                     String.format("Invalid partition: %d. Partition number should always be r
78         this.topic = topic;
79         this.partition = partition;
80         this.key = key;
81         this.value = value;
```

So what we can do is we can create a producer record object outside of send method.

1.25x 2:19 / 4:26

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

```java
    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        ProducerRecord<String, ProductCreatedEvent> record = new ProducerRecord<>(
                "product-created-events-topic",
                productId,
                productCreatedEvent);
        record.headers().add("messageId", UUID.randomUUID().toString().getBytes());

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send(record).get();

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());
        LOGGER.info("Offset: " + result.getRecordMetadata().offset());

        LOGGER.info("***** Returning product id");

        return productId;
    }

}
```

So I will use product ID as message key, and I will use message ID as another unique identifier that

106. Reading a unique id from message header

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

```java
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(@Payload ProductCreatedEvent productCreatedEvent,
            @Header("messageId") String messageId,
            @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch (HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch (Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

    }
```

So this will read a header with a name Kafka underscore received message key.

Pause 1.25x 3:10 / 4:07

106. Reading a unique id from message header

```java
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;

    public ProductCreatedEventHandler(RestTemplate restTemplate) {
        this.restTemplate = restTemplate;
    }

    @KafkaHandler
    public void handle(@Payload ProductCreatedEvent productCreatedEvent,
            @Header(value = "messageId", required=false) String messageId,
            @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch (HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch (Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

    }
```

So in this case, if message ID header is missing, then instead of getting exception, the value of

Pause 1.25x 3:36 / 4:07

107. Adding database-related dependencies

mvnrepository.com

MVN REPOSITORY

Search: H2

Found 449 results

Sort: relevance | popular

1. H2 Database Engine — 8,333 usages — MPL | EPL
com.h2database » h2
A fast SQL database that can run embedded or a server mode with support for transactions, encryption, full search, etc. Storage can be disk-based or in-memory.
Last Release on Sep 18, 2023

2. WSO2 Carbon Orbit H2 OSGi Console Service — 38 usages
org.wso2.orbit.com.h2database » h2
WSO2 Carbon Orbit H2 OSGi Console Service
Last Release on Feb 6, 2023

3. WSO2 Carbon Orbit H2 Osgi Console Service — 127 usages — MPL | EPL
com.h2database.wso2 » h2-database-engine
WSO2 Carbon Orbit H2 Osgi Console Service
Last Release on Aug 8, 2012

4. Database H2 — 89 usages — EPL
org.eclipse.dirigible » dirigible-database-h2
Database H2

5. H2 — 26 usages

The very first option at the top is what I'm looking for, H2 Database Engine.

1.25x 1:16 / 3:34

107. Adding database-related dependencies

Found 68821 results

1. Spring Boot Starter Data JPA — 2,388 usages
org.springframework.boot » spring-boot-starter-data-jpa
Starter for using Spring Data JPA with Hibernate
Last Release on Feb 22, 2024

2. Spring Boot Starter Test — 12,426 usages
org.springframework.boot » spring-boot-starter-test
Starter for testing Spring Boot applications with libraries including JUnit Jupiter, Hamcrest and Mockito
Last Release on Feb 22, 2024

3. Spring Boot Starter Web — 12,112 usages
org.springframework.boot » spring-boot-starter-web
Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container
Last Release on Feb 22, 2024

4. Spring Boot Starter — 7,902 usages
org.springframework.boot » spring-boot-starter
Core starter, including auto-configuration support, logging and YAML
Last Release on Feb 22, 2024

5. Spring Boot Starter Data Redis — 1,898 usages

And the very first item in the search result is what I'm looking for.

1.25x 2:29 / 3:34

107. Adding database-related dependencies

```
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.kafka</groupId>
        <artifactId>spring-kafka</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.kafka</groupId>
        <artifactId>spring-kafka-test</artifactId>
        <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.h2database/h2 -->
    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
    </dependency>
    <!--
    https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starte
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>


</dependencies>

<build>
```

All right, so now I can save changes to Pom.xml file and configure database access in Application properties

1.25x 3:26 / 3:34

108. Configure database connection details

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.s
consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdevelop

spring.datasource.username=sergey
spring.datasource.password=sergey
```

You can set any other username and password here.

1.25x 0:30 / 2:43

108. Configure database connection details

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.s
consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdevelop

spring.datasource.username=sergey
spring.datasource.password=sergey
```

```
spring.datasource.url=jdbc:h2:mem:testdb
```

This is a URL for connecting to database and the test db.

1.25x 0:59 / 2:43

108. Configure database connection details

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.s
consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdevelop

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
```

It should always be Augt dot driver.

1.25x 1:24 / 2:43

108. Configure database connection details

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092,localhost:9094
#spring.kafka.consumer.key-deserializer=org.apache.kafka.common.serialization.StringDeserializer
#spring.kafka.consumer.value-deserializer=org.springframework.kafka.support.serializer.JsonDeserializer
consumer.group-id=product-created-events
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.ws.core

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

1.25x 2:34 / 2:43

109. Creating JPA Entity

Java Class

Create a new Java class.

Source folder: EmailNotificationMicroservice/src/main/java

Package: com.appsdeveloperblog.ws.emailnotification.io

Enclosing type:

Name: ProcessedEventEntity

Modifiers: public package private protected abstract final static none sealed non-sealed final

Superclass: java.lang.Object

Interfaces: java.io.Serializable

Which method stubs would you like to create?

public static void main(String[] args)

Constructors from superclass

Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

Generate comments

Next I will click on finish button to create this new class.

Generate Getters and Setters

Select getters and setters to create:

serialVersionUID

id

messageId

productId

Select All

Deselect All

Select Getters

Select Setters

Allow setters for final fields (remove 'final' modifier from fields if necessary)

Insertion point: After 'productId'

Sort by: Fields in getter/setter pairs

Access modifier: public protected package private final synchronized

Generate method comments

The format of the getters/setters may be configured on the Code Templates preference page.

6 of 8 selected.

No need to generate getters and setters for Serialversionuid.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 62% Mon 26 Aug 11:55 AM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Thu Feb 29 3:10 PM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/io/ProcessedEventEntity.java - Spring Tool Suite 4

Package Explorer

- EmailNotificationMicroservice [boot] [EmailNotification
  - src/main/java
    - com.appsdeveloperblog.ws.emailnotification
    - com.appsdeveloperblog.ws.emailnotification.error
    - com.appsdeveloperblog.ws.emailnotification.handler
    - com.appsdeveloperblog.ws.emailnotification.io
      - ProcessedEventEntity.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
- ProductsMicroservice [boot] [ProductsMicroservice uniq

Boot Dashboard

- local

1 elements hidden by filter

*ProcessedEventEntity.java

```
1  import jakarta.persistence.Id;
2  import jakarta.persistence.Table;
10
11 @Entity
12 @Table(name="processed-events")
13 public class ProcessedEventEntity implements Serializable {
14
15     private static final long serialVersionUID = 3687553269742697084L;
16
17     @Id
18     @GeneratedValue
19     private long id;
20
21     @Column(nullable=false, unique=true)
22     private String messageId;
23
24     @Column(nullable=false)
25     private String productId;
26
27
28     public ProcessedEventEntity() {
29         super();
30     }
31
32     public ProcessedEventEntity(long id, String messageId, String productId) {
33         super();
34         this.id = id;
35         this.messageId = messageId;
36         this.productId = productId;
37     }
38
39     public long getId() {
40         return id;
41     }
42
43     public void setId(long id) {
44         this.id = id;
45     }
```

So I've generated two constructors and maybe delete this call to super.

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 64% Mon 26 Aug 11:56 AM

```
ventRepository.java ×
s.emailnotification.io;

.jpa.repository.JpaRepository;

Repository extends JpaRepository<ProcessedEventEntity, Long> {
```

Now this part that we have here, it is called generic type parameter and it tells GPU repository interface

Udemy

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/io/ProcessedEventRepository.java - Spring Tool Suite 4

ProcessedEventEntity.java | *ProcessedEventRepository.java

```java
package com.appsdeveloperblog.ws.emailnotification.io;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ProcessedEventRepository extends JpaRepository<ProcessedEventEntity, Long> {

}
```

So this means that my GPA repository will be able to perform operations like save processed event entity

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/io/ProcessedEventRepository.java - Spring Tool Suite 4

ProcessedEventEntity.java | *ProcessedEventRepository.java

```java
package com.appsdeveloperblog.ws.emailnotification.io;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface ProcessedEventRepository extends JpaRepository<ProcessedEventEntity, Long> {

}
```

Now using this repository annotation is optional in this case, because Spring framework will be able

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/io/ProcessedEventEntity.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java | *ProcessedEventEntity.java

```java
package com.appsdeveloperblog.ws.emailnotification.io;

import java.io.Serializable;

@Entity
@Table(name="processed-events")
public class ProcessedEventEntity implements Serializable {

    private static final long serialVersionUID = 3687553269742697084L;

    @Id
    @GeneratedValue
    private long id;

    @Column(nullable=false, unique=true)
    private String messageId;

    @Column(nullable=false)
    private String productId;


    public ProcessedEventEntity() {
    }

    public ProcessedEventEntity(String messageId, String productId) {
        this.messageId = messageId;
        this.productId = productId;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }
```

I don't need to add database ID into constructor for this entity, so I will remove it.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java | ProcessedEventEntity.java

```java
        this.restTemplate = restTemplate;
        this.processedEventRepository = processedEventRepository;
    }

    @KafkaHandler
    public void handle(@Payload ProductCreatedEvent productCreatedEvent,
            @Header("messageId") String messageId,
            @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch (HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch (Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

        // Save a unique message id in a database table
        processedEventRepository.save(new ProcessedEventEntity(messageId, productCreatedEvent.getProductId()));

    }

}
```

exists in a database, and as a result we will get exception.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 71% Mon 26 Aug 12:02 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Mar 1 10:17 AM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java × ProcessedEventEntity.java

```
    @KafkaHandler
    public void handle(@Payload ProductCreatedEvent productCreatedEvent, @Header("messageId") String messageId,
            @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

            if (response.getStatusCode().value() == HttpStatus.OK.value()) {
                LOGGER.info("Received response from a remote service: " + response.getBody());
            }
        } catch (ResourceAccessException ex) {
            LOGGER.error(ex.getMessage());
            throw new RetryableException(ex);
        } catch (HttpServerErrorException ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        } catch (Exception ex) {
            LOGGER.error(ex.getMessage());
            throw new NotRetryableException(ex);
        }

        // Save a unique message id in a database table
        try {
            processedEventRepository.save(new ProcessedEventEntity(messageId, productCreatedEvent.getProductId()));
        } catch (DataIntegrityViolationException ex) {
            throw new NotRetryableException(ex);
        }

    }

}
```

Now because I'm performing a modified operation here, I'm saving information in a database table.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 71% Mon 26 Aug 12:02 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Mar 1 10:18 AM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

@Transactional

```
    @Transactional
```

Transactional cannot be resolved to a type

6 quick fixes available:

- Import 'Transactional' (jakarta.transaction)
- Import 'Transactional' (org.springframework.transaction.annotation)
- Create annotation 'Transactional'
- Change to 'TransactionalEventListener' (org.springframework.transaction.event)
- Change to 'TransactionScoped' (jakarta.transaction)
- Fix project setup...

Let's import transactional annotation from org spring framework.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/io/ProcessedEventRepository.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java | ProcessedEventEntity.java | *ProcessedEventRepository.java

```java
package com.appsdeveloperblog.ws.emailnotification.io;

import org.springframework.data.jpa.repository.JpaRepository;

@Repository
public interface ProcessedEventRepository extends JpaRepository<ProcessedEventEntity, Long> {

    ProcessedEventEntity findByMessageId(String messageId);

}
```

We can now use this method to look up records in a database table by message ID.

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

*ProductCreatedEventHandler.java | ProcessedEventEntity.java | ProcessedEventRepository.java

```java
package com.appsdeveloperblog.ws.emailnotification.handler;

import org.slf4j.Logger;

@Component
@KafkaListener(topics = "product-created-events-topic")
public class ProductCreatedEventHandler {

    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private RestTemplate restTemplate;
    private ProcessedEventRepository processedEventRepository;

    public ProductCreatedEventHandler(RestTemplate restTemplate, ProcessedEventRepository processedEventRepository) {
        this.restTemplate = restTemplate;
        this.processedEventRepository = processedEventRepository;
    }

    @Transactional
    @KafkaHandler
    public void handle(@Payload ProductCreatedEvent productCreatedEvent, @Header("messageId") String messageId,
            @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
        LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
                + productCreatedEvent.getProductId());

        // Check if this message was already processed before

        ProcessedEventEntity existingRecord = processedEventRepository.findByMessageId(messageId);

        if(existingRecord!=null) {
            LOGGER.info("Found a duplicate message id: {}", existingRecord.getMessageId());
            return;
        }

        String requestUrl = "http://localhost:8082/response/200";

        try {
            ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);
```

And this is it.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 77% Mon 26 Aug 12:08 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Mar 1 6:06 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEventHandler.java | *ProductServiceImpl.java

```java
    public ProductServiceImpl(KafkaTemplate<String, ProductCreatedEvent> kafkaTemplate) {
        this.kafkaTemplate = kafkaTemplate;
    }

    @Override
    public String createProduct(CreateProductRestModel productRestModel) throws Exception {

        String productId = UUID.randomUUID().toString();

        // TODO: Persist Product Details into database table before publishing an Event

        ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
                productRestModel.getTitle(), productRestModel.getPrice(),
                productRestModel.getQuantity());

        LOGGER.info("Before publishing a ProductCreatedEvent");

        ProducerRecord<String, ProductCreatedEvent> record = new ProducerRecord<>(
                "product-created-events-topic",
                productId,
                productCreatedEvent);
        record.headers().add("messageId", "123".getBytes());

        SendResult<String, ProductCreatedEvent> result =
                kafkaTemplate.send(record).get();

        LOGGER.info("Partition: " + result.getRecordMetadata().partition());
        LOGGER.info("Topic: " + result.getRecordMetadata().topic());
        LOGGER.info("Offset: " + result.getRecordMetadata().offset());

        LOGGER.info("***** Returning product id");

        return productId;
    }

}
```

So I will remove this random Uuid for now, and I will hardcode something like 1 to 3 when the consumer

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 78% Mon 26 Aug 12:09 PM

SpringToolSuite4 Fri Mar 1 6:07 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEventHandler.java | ProductServiceImpl.java

Proble... Javadoc Declar... Console Progress Debug

```
2024-03-01T18:04:22.700-05:00 INFO 23363 --- [
2024-03-01T18:04:22.720-05:00 INFO 23363 --- [
2024-03-01T18:04:23.026-05:00 INFO 23363 --- [ntainer#0-
2024-03-01T18:04:23.027-05:00 INFO 23363 --- [ntainer#0-
2024-03-01T18:04:23.030-05:00 INFO 23363 --- [ntainer#0-
2024-03-01T18:04:23.049-05:00 INFO 23363 --- [ntainer#0-
2024-03-01T18:04:23.050-05:00 INFO 23363 --- [ntainer#0-
```

Confirm Perspective Switch

This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective supports application debugging by providing views for displaying the debug stack, variables and breakpoints.

Switch to this perspective?

Remember my decision

No Switch

Udemy

113. Trying how it works

Final chance to help improve integrations with Postman! Take the survey now to help us better understand your toolset and priorities. We will close the survey on Friday, January 12.

POST http://localhost:63057/products

```
{
    "title":"iPad Pro",
    "price":1250,
    "quantity":19
}
```

Status: 201 Created Time: 30 ms Size: 205 B

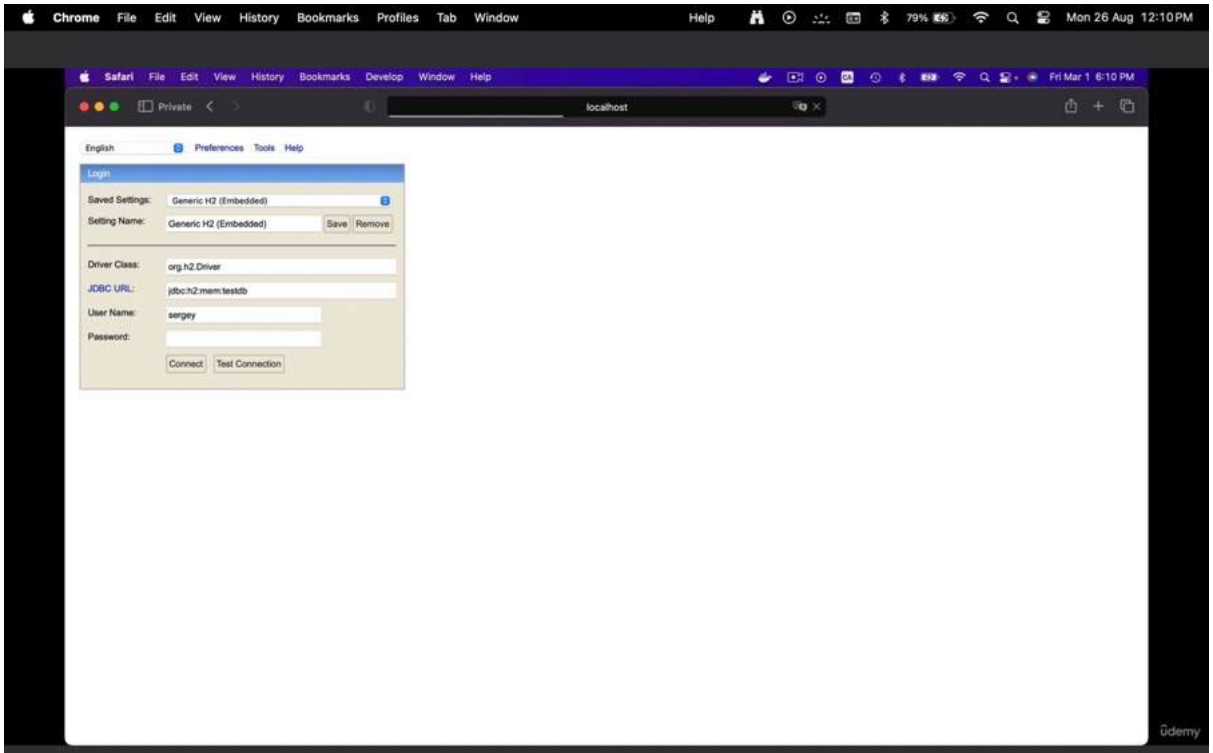
```
594e0136-d75b-4b3e-a3c7-c21f5da388d8
```

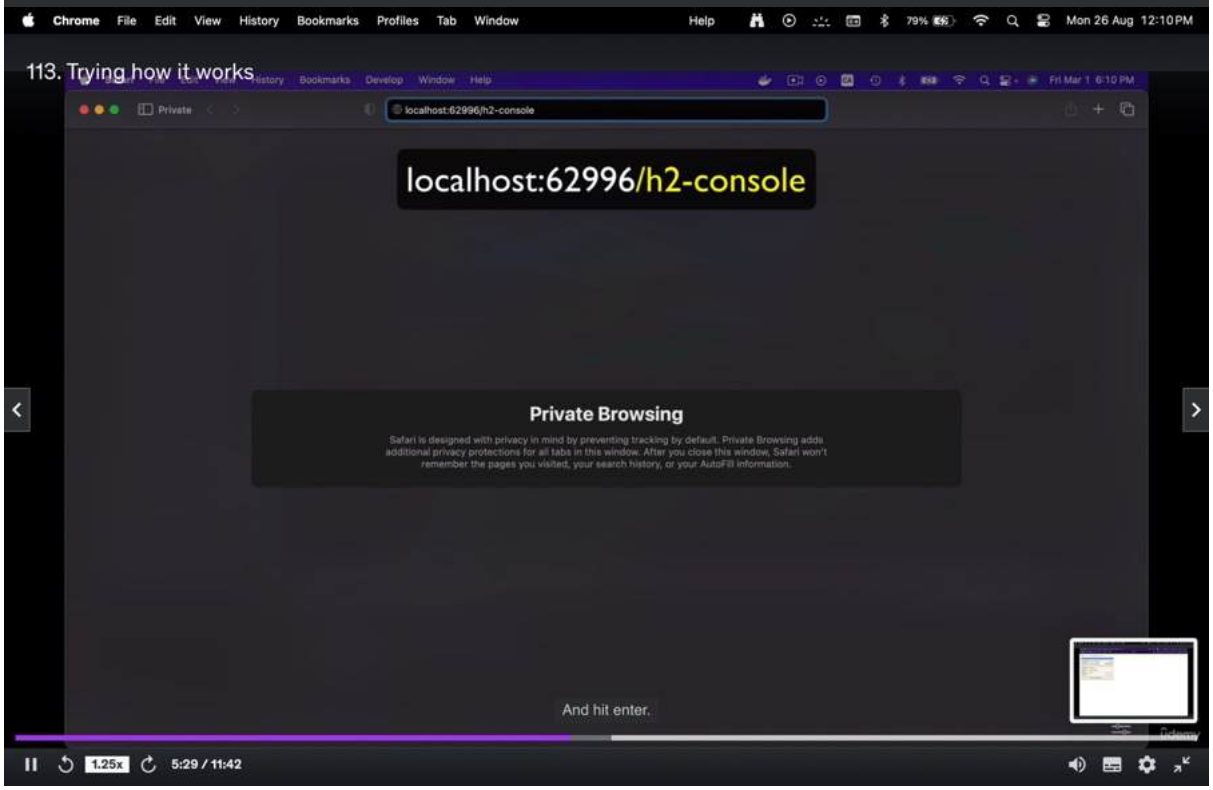
So now that I have updated port number, I can send this Http request.

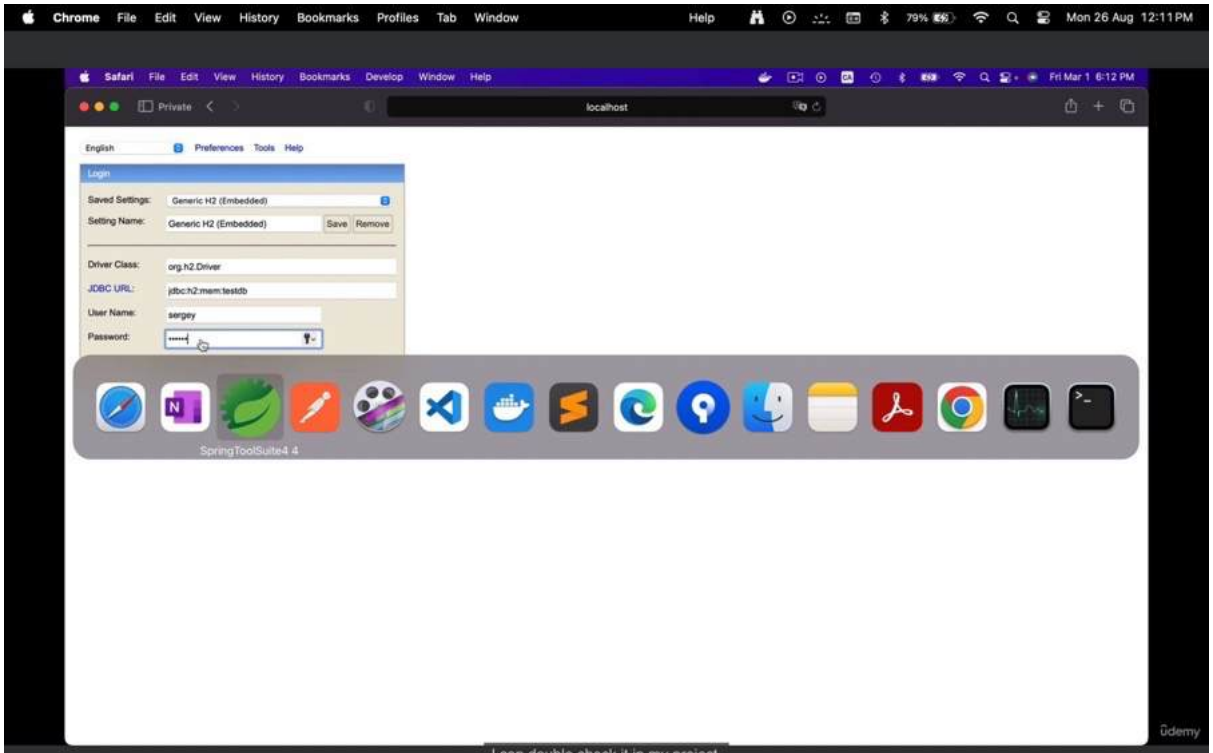
1.25x 3:56 / 11:42

```
2024-03-01T18:09:23.543-05:00  INFO 23363 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator : [Con
2024-03-01T18:09:23.543-05:00  INFO 23363 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoordinator : [Con
2024-03-01T18:09:23.570-05:00  INFO 23363 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionState   : [Con
2024-03-01T18:09:24.022-05:00  INFO 23363 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.SubscriptionState   : [Con
2024-03-01T18:09:24.025-05:00  INFO 23363 --- [ntainer#0-0-C-1] o.s.k.l.KafkaMessageListenerContainer   : proc
```

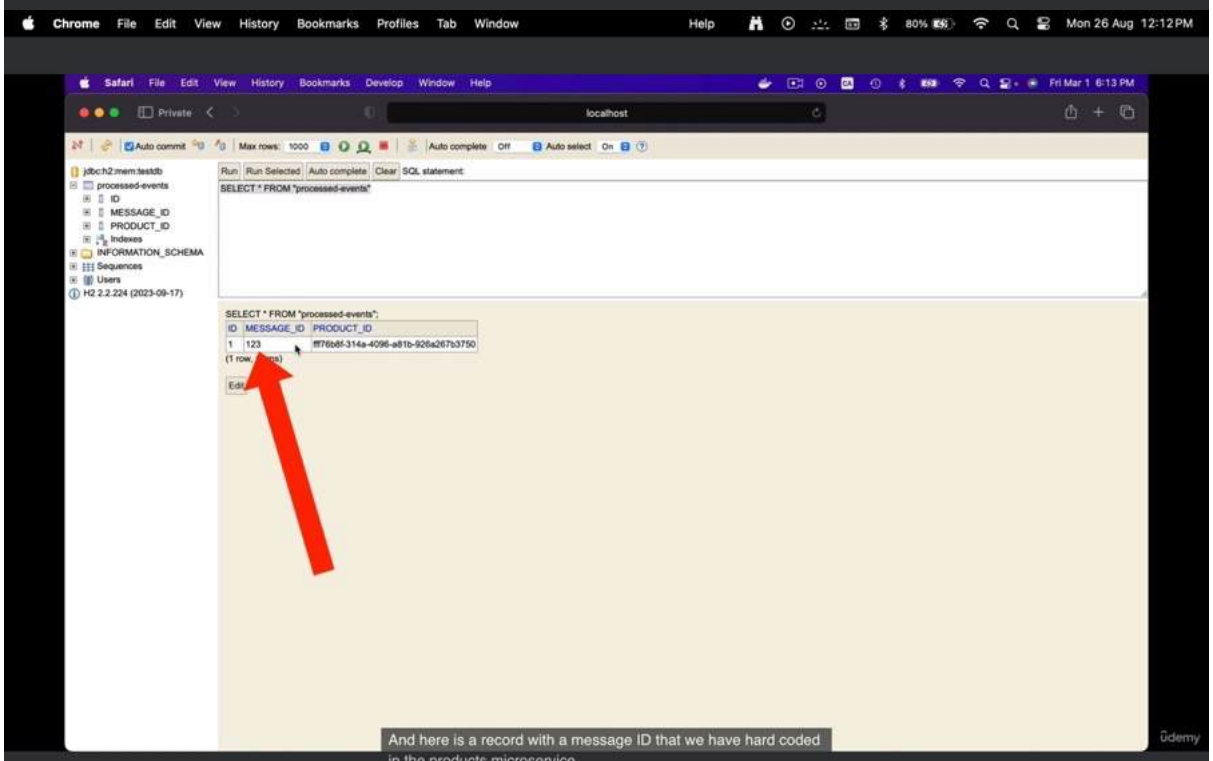
H2 provides us with a very nice web interface.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 26 Aug 12:10 PM
Safari File Edit View History Bookmarks Develop Window Help
Fri Mar 1 6:10 PM
Private
localhost
English
Preferences Tools Help
Login
Saved Settings:
Generic H2 (Embedded)
Setting Name:
Generic H2 (Embedded)
Save
Remove
Driver Class:
org.h2.Driver
JDBC URL:
jdbc:h2:mem:testdb
User Name:
sergey
Password:
Connect
Test Connection
Udemy


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 26 Aug 12:10 PM
113. Trying how it works
Private
localhost:62996/h2-console
localhost:62996/h2-console
Private Browsing
Safari is designed with privacy in mind by preventing tracking by default. Private Browsing adds additional privacy protections for all tabs in this window. After you close this window, Safari won't remember the pages you visited, your search history, or your AutoFill information.
And hit enter.
1.25x
5:29 / 11:42



I can double check it in my project.



And here is a record with a message ID that we have hard coded in the products microservice.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 80% Mon 26 Aug 12:12 PM

Postman File Edit View Window Help Fri Mar 1 6:14 PM

Home Workspaces API Network Upgrade

Final chance to help improve integrations with Postman! Take the survey now to help us better understand your toolset and priorities. We will close the survey on Friday, January 12. Take Survey Now

POST http://localhost:6305 No Environment

http://localhost:63057/products Save

POST http://localhost:63057/products Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "title":"iPad Pro",
3     "price":1250,
4     "quantity":19
5 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 524 ms Size: 205 B Save as example

Pretty Raw Preview Visualize Text

```
1 fff76b8f-314a-4096-a81b-926a267b3750
```

Online Find and replace Console Runner Start Proxy Cookies Trash

To do that I will go to postman and I will simply repeat this Http request.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 80% Mon 26 Aug 12:12 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Mar 1 6:14 PM

Event-Driven Microservices with Apache Kafka - EmailNotificationMicroservice/src/main/java/com/appsdeveloperblog/ws/emailnotification/handler/ProductCreatedEventHandler.java - Spring Tool Suite 4

ProductCreatedEventHandler.java ProductServiceImpl.java application.properties

```
37     }
38
39     @Transactional
40     @KafkaHandler
41     public void handle(@Payload ProductCreatedEvent productCreatedEvent, @Header("messageId") String mess
42             @Header(KafkaHeaders.RECEIVED_KEY) String messageKey) {
43         LOGGER.info("Received a new event: " + productCreatedEvent.getTitle() + " with productId: "
44                 + productCreatedEvent.getProductId());
45
46         // Check if this message was already processed before
47         ProcessedEventEntity existingRecord = processedEventRepository.findByMessageId(messageId);
48
49         if(existingRecord!=null) {
50             LOGGER.info("Found a duplicate message id: {}", existingRecord.getMessageId());
51             return;
52         }
53
54         String requestUrl = "http://localhost:8082/response/200";
55
56         try {
57             ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, Str
58
59             if (response.getStatusCode().value() == HttpStatus.OK.value()) {
60                 LOGGER.info("Received response from a remote service: " + response.getBody());
61             }
62         } catch (ResourceAccessException ex) {
63             LOGGER.error(ex.getMessage());
```

Console Problems Debug Shell Boot Dashboard

```
2024-03-01T18:10:50.889-05:00  INFO 23363 --- [o-auto-1-exec-6] o.a.c.c.C.[Tomcat].[localhost].[/]       : Init
2024-03-01T18:10:50.889-05:00  INFO 23363 --- [o-auto-1-exec-6] o.s.web.servlet.DispatcherServlet        : Init
2024-03-01T18:10:50.896-05:00  INFO 23363 --- [o-auto-1-exec-6] o.s.web.servlet.DispatcherServlet        : Comp
2024-03-01T18:13:23.146-05:00  INFO 23363 --- [ntainer#0-0-C-1] org.apache.kafka.clients.NetworkClient   : [Con
2024-03-01T18:14:24.243-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Rece
```

And this time I have a debugger breakpoint triggered inside of if condition.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 81% Mon 26 Aug 12:13 PM

Safari File Edit View History Bookmarks Develop Window Help Fri Mar 1 6:15 PM

Private localhost

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:testdb
processed-events
ID
MESSAGE_ID
PRODUCT_ID
Indexes
INFORMATION_SCHEMA
Sequences
Users
H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM "processed-events"
```

SELECT * FROM "processed-events";

ID	MESSAGE_ID	PRODUCT_ID
1	123	ffff76b8f-314a-4096-a81b-926a267b3750

(1 row, 0 ms)

Edit

Udemy

And we don't have any duplicate message IDs here.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 82% Mon 26 Aug 12:14 PM

SpringToolSuite4 File Edit Source Refactor Source Navigate Search Project Run Window Help Fri Mar 1 6:19 PM

Event-Driven Microservices with Apache Kafka - ProductsMicroservice/src/main/java/com/appsdeveloperblog/ws/products/service/ProductServiceImpl.java - Spring Tool Suite 4

ProductCreatedEventHandler.java | *ProductServiceImpl.java | application.properties

```
22      }
23
24      @Override
25      public String createProduct(CreateProductRestModel productRestModel) throws Exception {
26
27          String productId = UUID.randomUUID().toString();
28
29          // TODO: Persist Product Details into database table before publishing an Event
30
31          ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
32                  productRestModel.getTitle(), productRestModel.getPrice(),
33                  productRestModel.getQuantity());
34
35          LOGGER.info("Before publishing a ProductCreatedEvent");
36
37          ProducerRecord<String, ProductCreatedEvent> record = new ProducerRecord<>(
38                  "product-created-events-topic",
39                  productId,
40                  productCreatedEvent);
41          record.headers().add("messageId", UUID.randomUUID().toString().getBytes());
42
43          SendResult<String, ProductCreatedEvent> result =
44                  kafkaTemplate.send(record).get();
45
46          LOGGER.info("Partition: " + result.getRecordMetadata().partition());
47          LOGGER.info("Topic: " + result.getRecordMetadata().topic());
48          LOGGER.info("Offset: " + result.getRecordMetadata().offset());
```

Console | Problems | Debug Shell | Boot Dashboard

```
2024-03-01T18:17:25.213-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Four
2024-03-01T18:17:26.019-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Rece
2024-03-01T18:17:26.023-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Four
2024-03-01T18:17:26.981-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Rece
2024-03-01T18:17:26.984-05:00  INFO 23363 --- [ntainer#0-0-C-1] c.a.w.e.h.ProductCreatedEventHandler     : Four
```

Udemy

If I run products microservice with this change then it should still work well.

```
}

@Override
public String createProduct(CreateProductRestModel productRestModel) throws Exception {

    String productId = UUID.randomUUID().toString();

    // TODO: Persist Product Details into database table before publishing an Event

    ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
            productRestModel.getTitle(), productRestModel.getPrice(),
            productRestModel.getQuantity());

    LOGGER.info("Before publishing a ProductCreatedEvent");

    ProducerRecord<String, ProductCreatedEvent> record = new ProducerRecord<>(
            "product-created-events-topic",
            productId,
            productCreatedEvent);
    record.headers().add("messageId", UUID.randomUUID().toString().getBytes());

    SendResult<String, ProductCreatedEvent> result =
            kafkaTemplate.send(record).get();

    LOGGER.info("Partition: " + result.getRecordMetadata().partition());
    LOGGER.info("Topic: " + result.getRecordMetadata().topic());
    LOGGER.info("Offset: " + result.getRecordMetadata().offset());
```

The new port number is 63191 will go back to postman.


```
}

@Override
public String createProduct(CreateProductRestModel productRestModel) throws Exception {

    String productId = UUID.randomUUID().toString();

    // TODO: Persist Product Details into database table before publishing an Event

    ProductCreatedEvent productCreatedEvent = new ProductCreatedEvent(productId,
            productRestModel.getTitle(), productRestModel.getPrice(),
            productRestModel.getQuantity());
```

The new port number is 63191 will go back to postman.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 82% Mon 26 Aug 12:14 PM

Postman File Edit View Window Help Fri Mar 1 6:20 PM

Home Workspaces API Network Upgrade

Final chance to help improve integrations with Postman! Take the survey now to help us better understand your toolset and priorities. We will close the survey on Friday, January 12. Take Survey Now

POST http://localhost:63191 No Environment

Collections APIs Environments History

http://localhost:63191/products Save

POST http://localhost:63191/products Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "title":"iPad Pro",
3     "price":1250,
4     "quantity":19
5 }
```

Body Cookies Headers (5) Test Results Status: 201 Created Time: 20 ms Size: 205 B Save as example

Pretty Raw Preview Visualize Text

```
1 7e362cd7-7edb-40aa-8829-8c06d7b10a52
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash

And send the Http request to create a new product.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 82% Mon 26 Aug 12:15 PM

Safari File Edit View History Bookmarks Develop Window Help Fri Mar 1 6:22 PM

Private localhost

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:testdb
- processed-events
  - ID
  - MESSAGE_ID
  - PRODUCT_ID
  - Indexes
- INFORMATION_SCHEMA
- Sequences
- Users

H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM "processed-events"
```

SELECT * FROM "processed-events";

ID	MESSAGE_ID	PRODUCT_ID
1	123	ff76b8f-314a-4096-a81b-926a267b3750
2	194a2517-520c-4184-91c3-3fed08157224	9ef142b5-3ed6-4c02-b5a5-a72ad6002e58
3	062de27d-7d18-40eb-a22b-6561e98137be	cb098993-abe7-416f-a3bc-7f87ff48382e
4	5cbbcc19-b1c7-4997-bc91-154b4c439767	7403c597-6db9-49e0-8e5b-e7d69a783072
5	0536a58d-9283-443f-9bd8-1b169e1a086f	c6eface8-9736-4c1f-8c44-c56cea948509
6	17f7b62a-e02d-46f6-97da-4fa41a7ed9d8	5d79c052-f185-4aae-8e63-d5649d1ddb06
7	3ea1e2ce-3ca0-4f0d-926a-7985379c0d08	b8025362-c902-4642-bdca-8f530ac0b6ce
8	500b4443-7c1a-4b2e-9b38-85547f1b506d	033f9144-0f7a-4097-a8c8-633af9900a9f

(8 rows, 1 ms)

Edit

Now all unique message IDs were successfully stored.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 83% Mon 26 Aug 12:16 PM

114. Introduction to Transactions in Apache Kafka

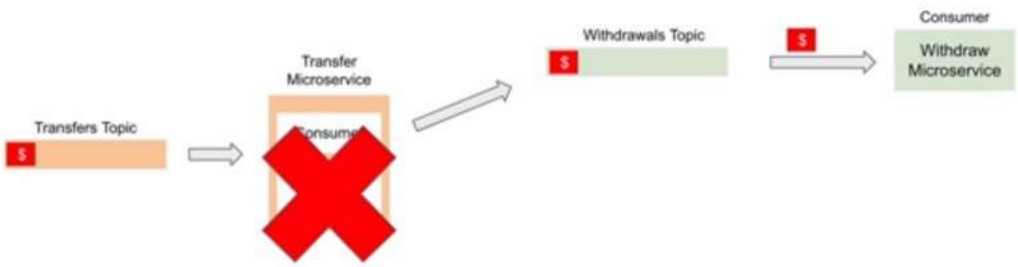
Why use transactions in Apache Kafka?

- All or nothing
- Exactly once
  - Idempotent Producer,
  - Transactions

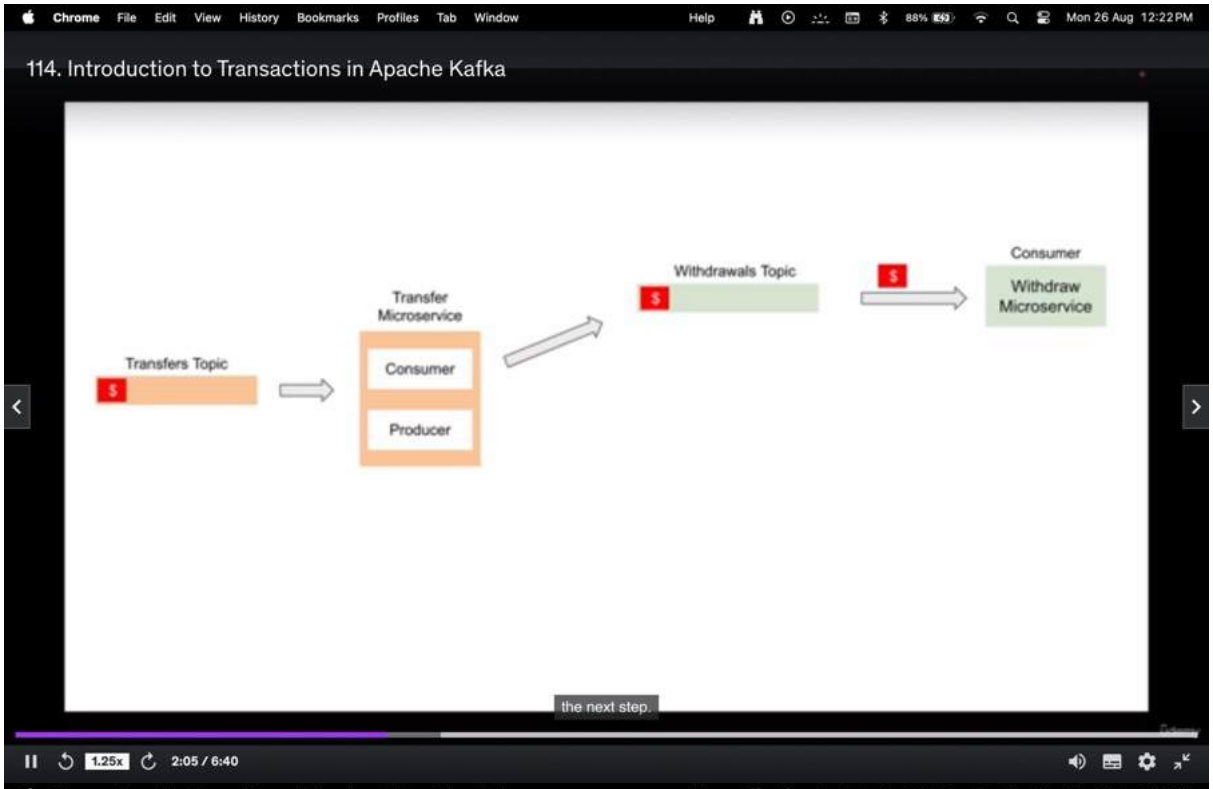
that consumer does not read message that is part of incomplete transaction.

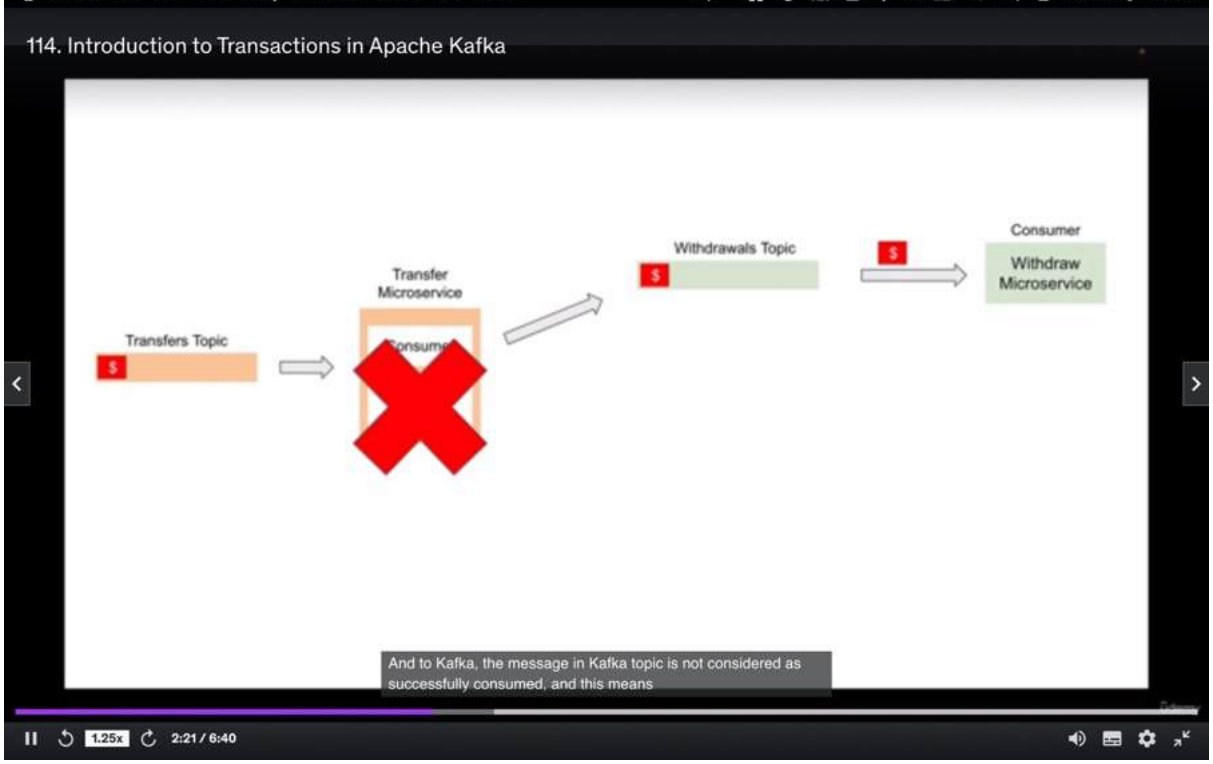
1.25x 0:53 / 6:40

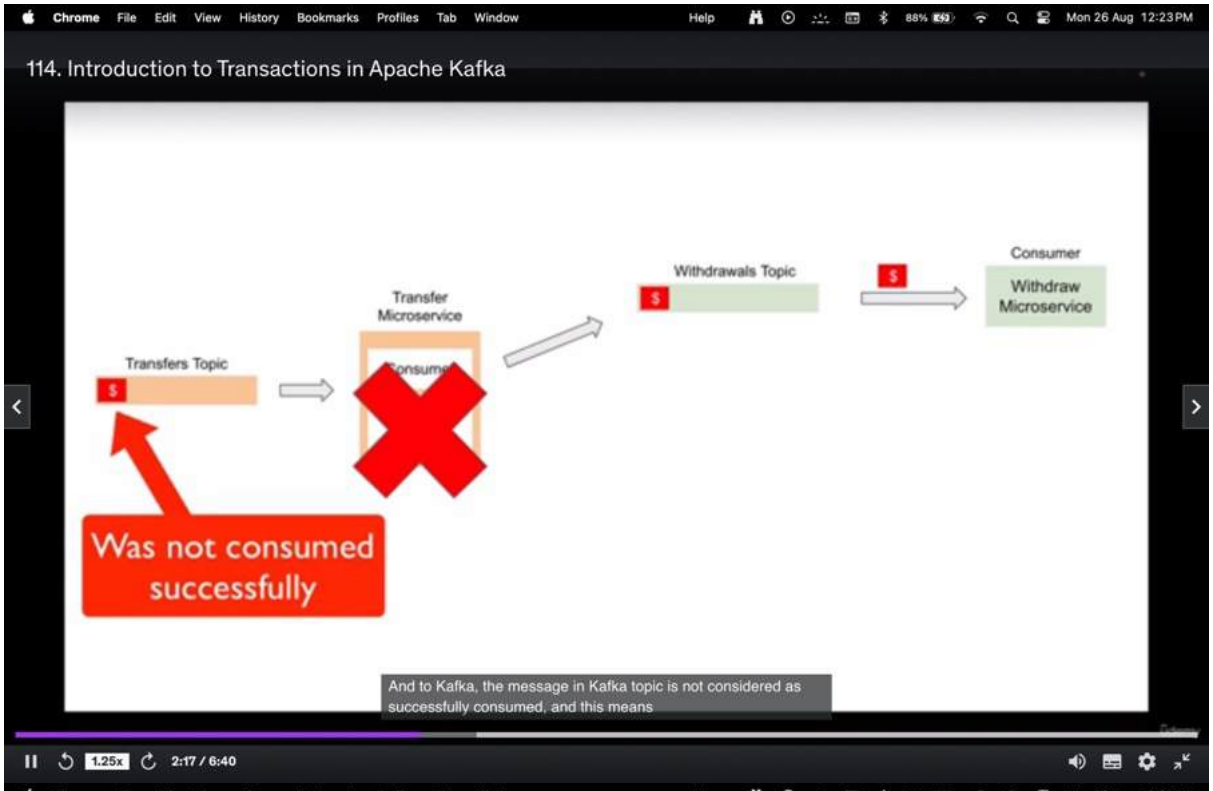
Chrome File Edit View History Bookmarks Profiles Tab Window Help 88% Mon 26 Aug 12:22 PM

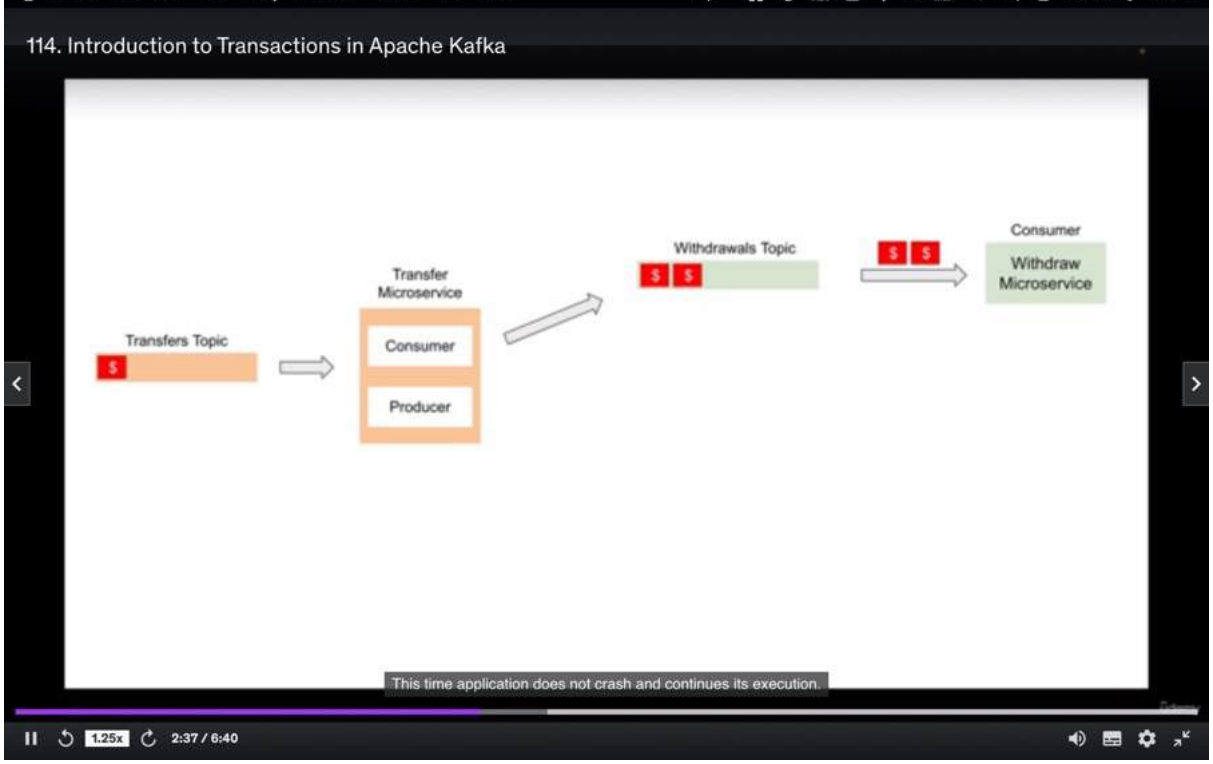


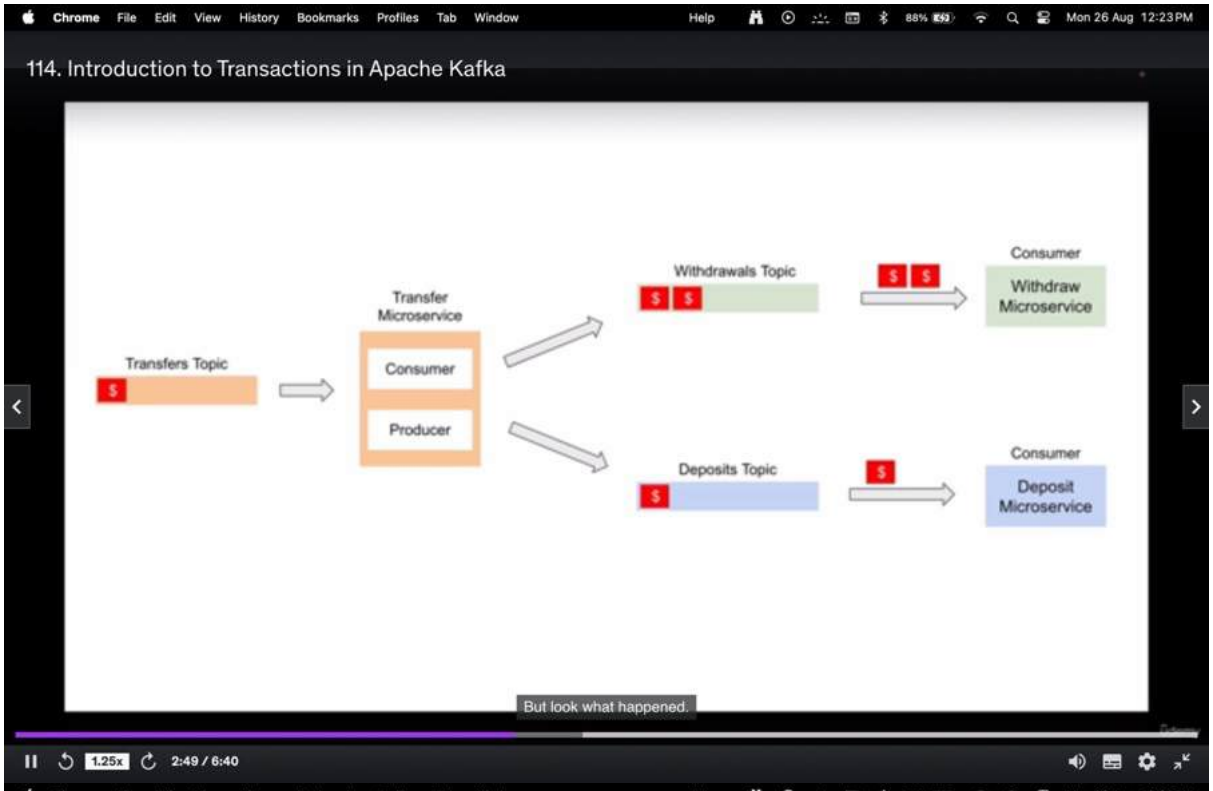
And to Kafka, the message in Kafka topic is not considered as successfully consumed, and this means

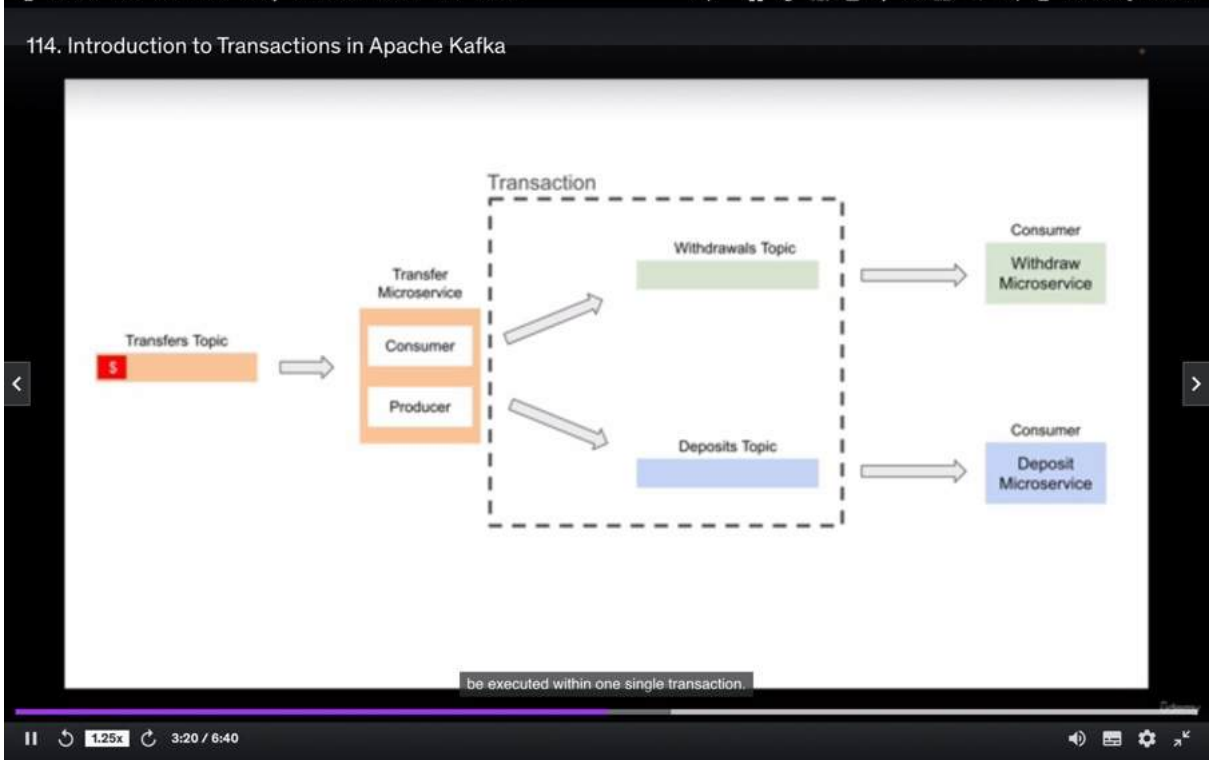

Chrome File Edit View History Bookmarks Profiles Tab Window
Help
Mon 26 Aug 12:22 PM
114. Introduction to Transactions in Apache Kafka
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Consumer
Withdraw Microservice
the next step.
1.25x
2:05 / 6:40

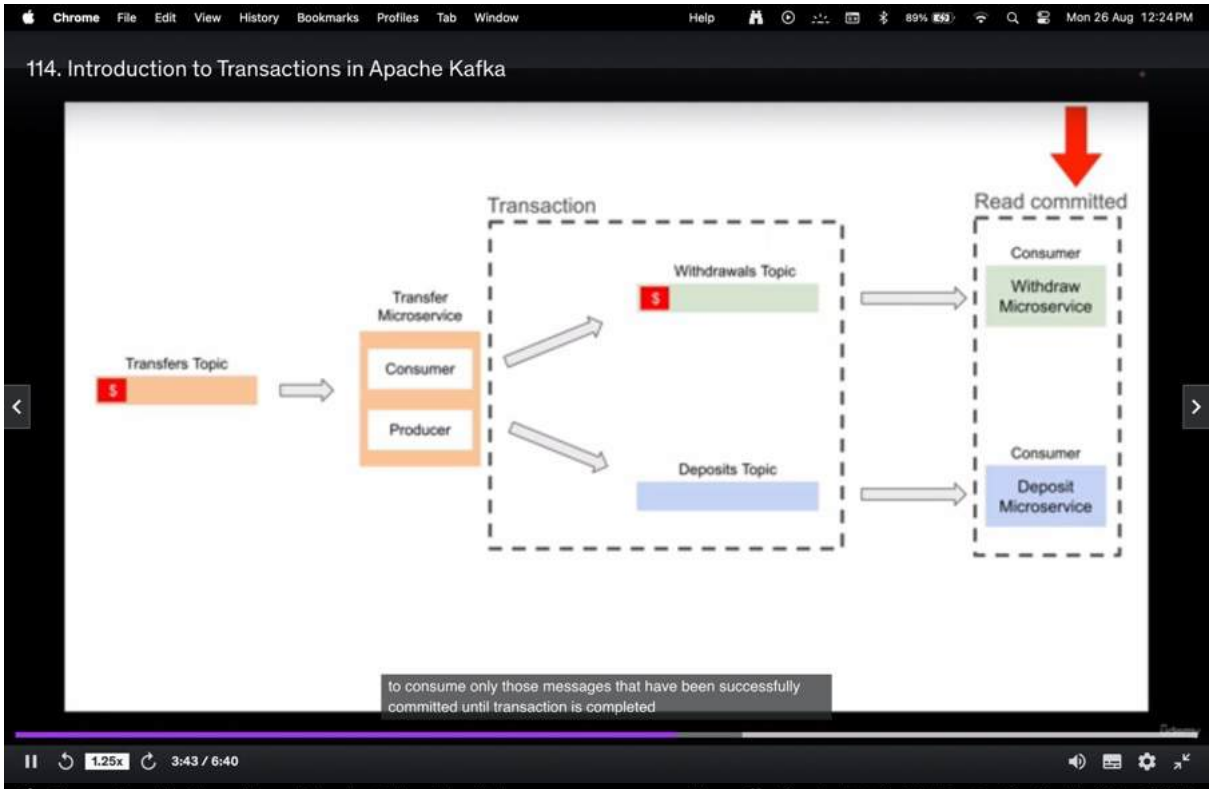

Chrome File Edit View History Bookmarks Profiles Tab Window
Help
Mon 26 Aug 12:22 PM
114. Introduction to Transactions in Apache Kafka
Transfers Topic
Transfer Microservice
Withdrawals Topic
Consumer
Withdraw Microservice
And to Kafka, the message in Kafka topic is not considered as successfully consumed, and this means
1.25x
2:21 / 6:40

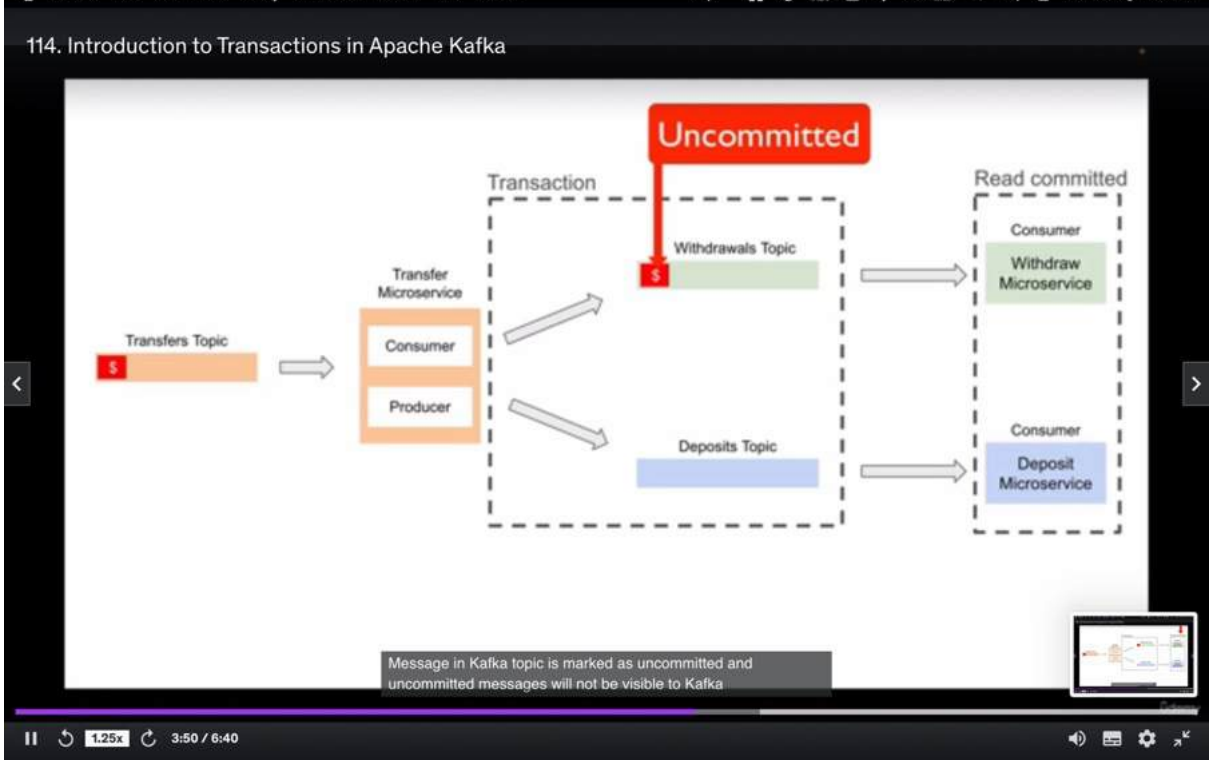

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 12:23 PM
114. Introduction to Transactions in Apache Kafka
Transfers Topic
Transfer Microservice
Withdrawals Topic
Consumer
Withdraw Microservice
Was not consumed successfully
And to Kafka, the message in Kafka topic is not considered as successfully consumed, and this means
1.25x 2:17 / 6:40

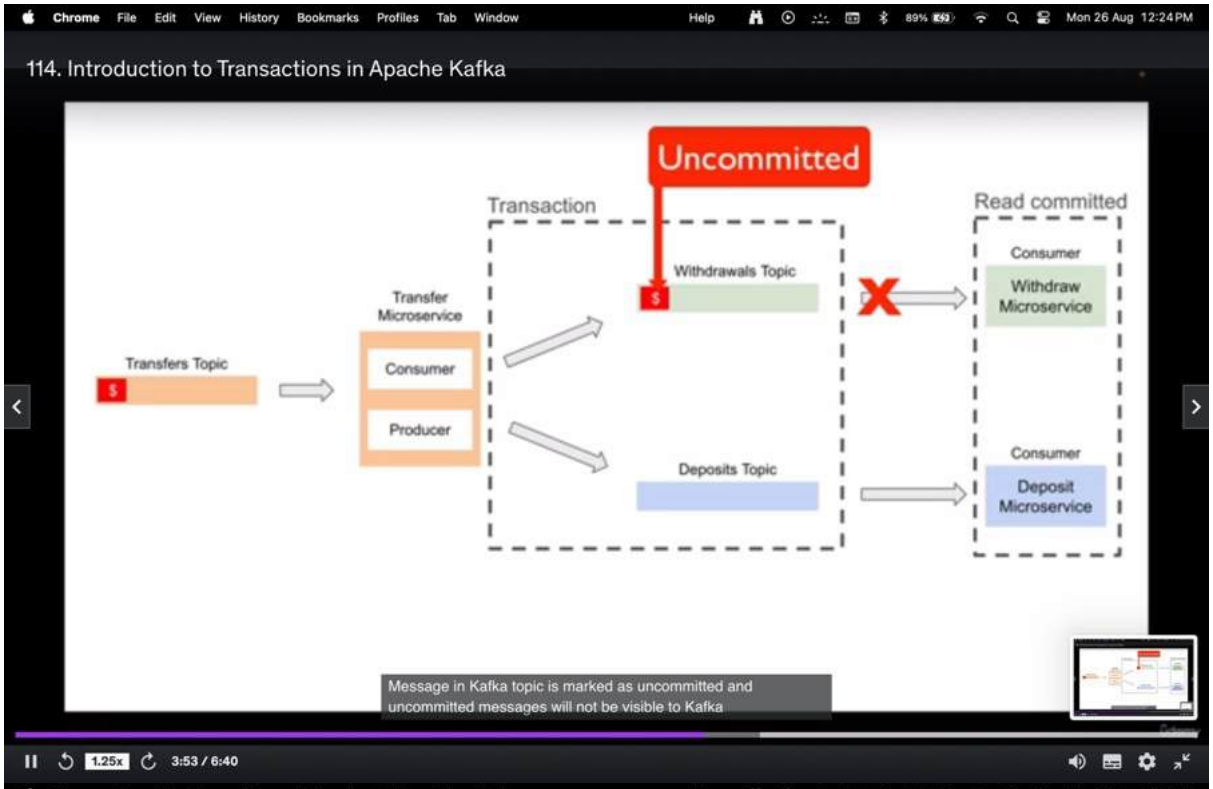

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 12:23 PM
114. Introduction to Transactions in Apache Kafka
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Consumer
Withdraw Microservice
This time application does not crash and continues its execution.
1.25x 2:37 / 6:40

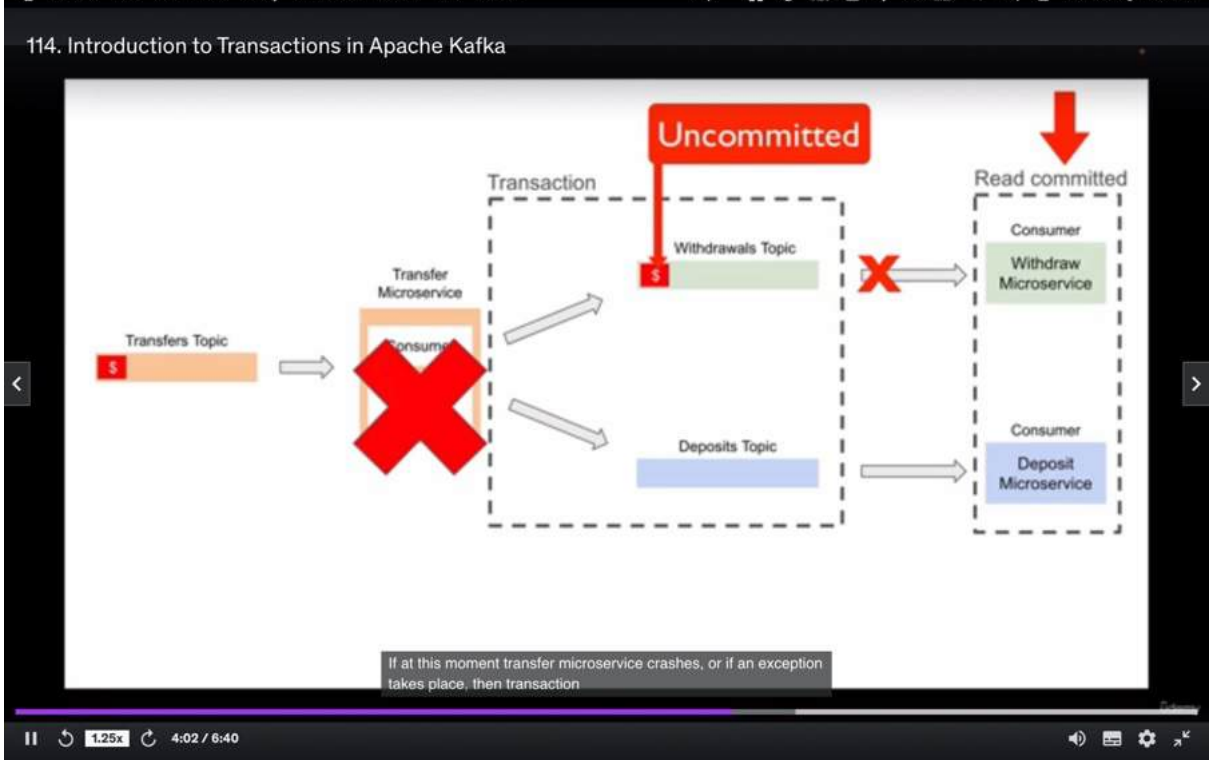

Chrome File Edit View History Bookmarks Profiles Tab Window
Help
Mon 26 Aug 12:23 PM
114. Introduction to Transactions in Apache Kafka
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
But look what happened.
1.25x
2:49 / 6:40

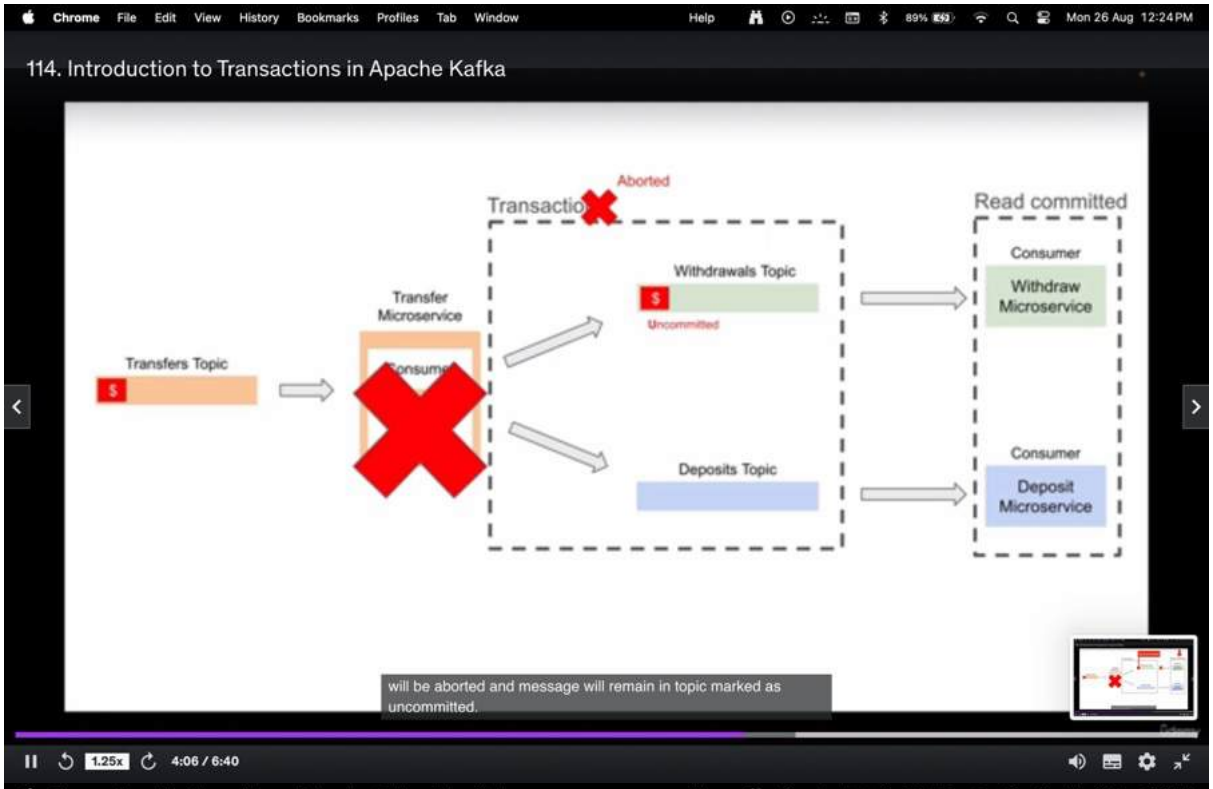

Chrome File Edit View History Bookmarks Profiles Tab Window
Help
Mon 26 Aug 12:23 PM
114. Introduction to Transactions in Apache Kafka
Transaction
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
be executed within one single transaction.
1.25x
3:20 / 6:40

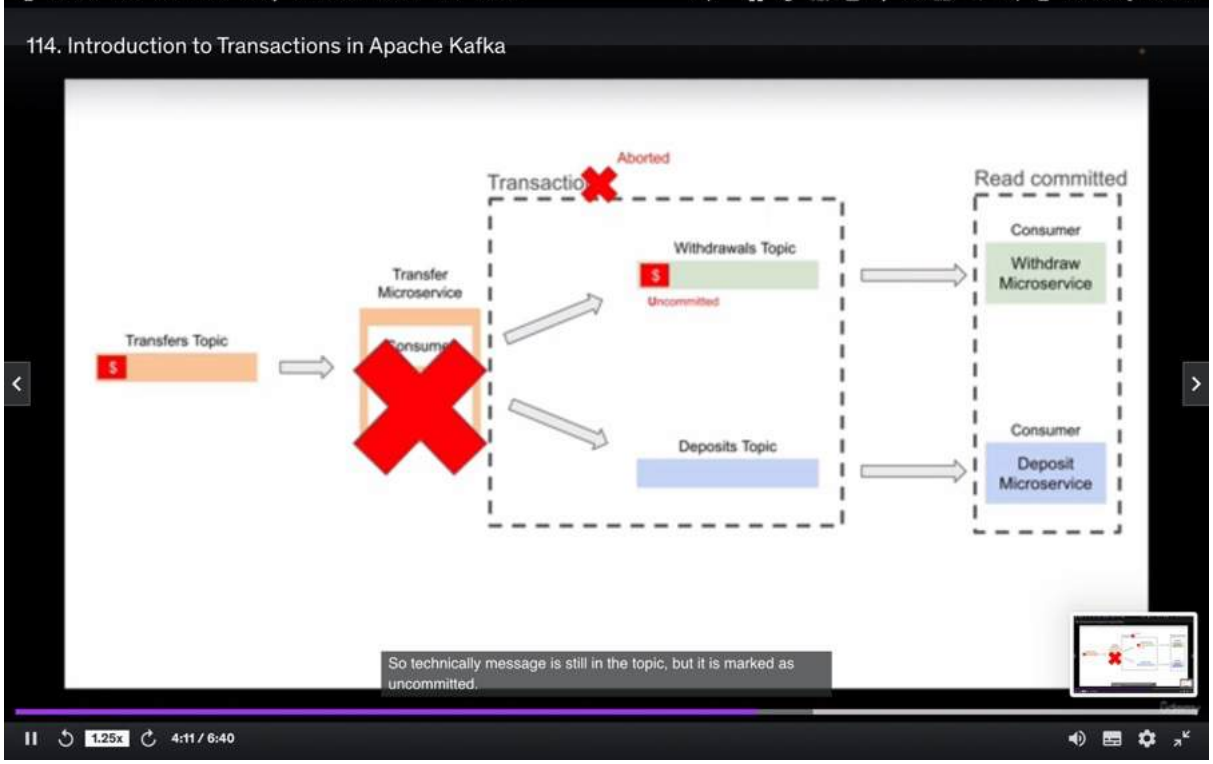

Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Transaction
Read committed
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
to consume only those messages that have been successfully committed until transaction is completed
1.25x 3:43 / 6:40

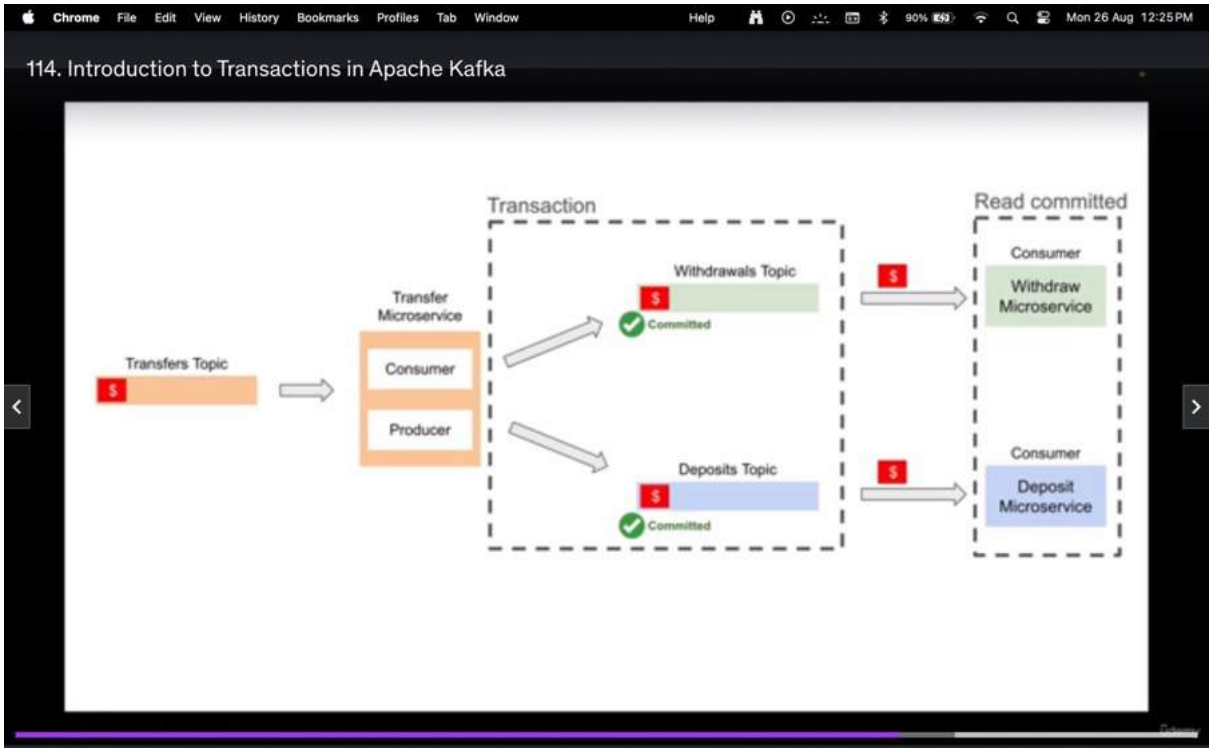

Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Uncommitted
Transaction
Read committed
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
Message in Kafka topic is marked as uncommitted and uncommitted messages will not be visible to Kafka
1.25x 3:50 / 6:40

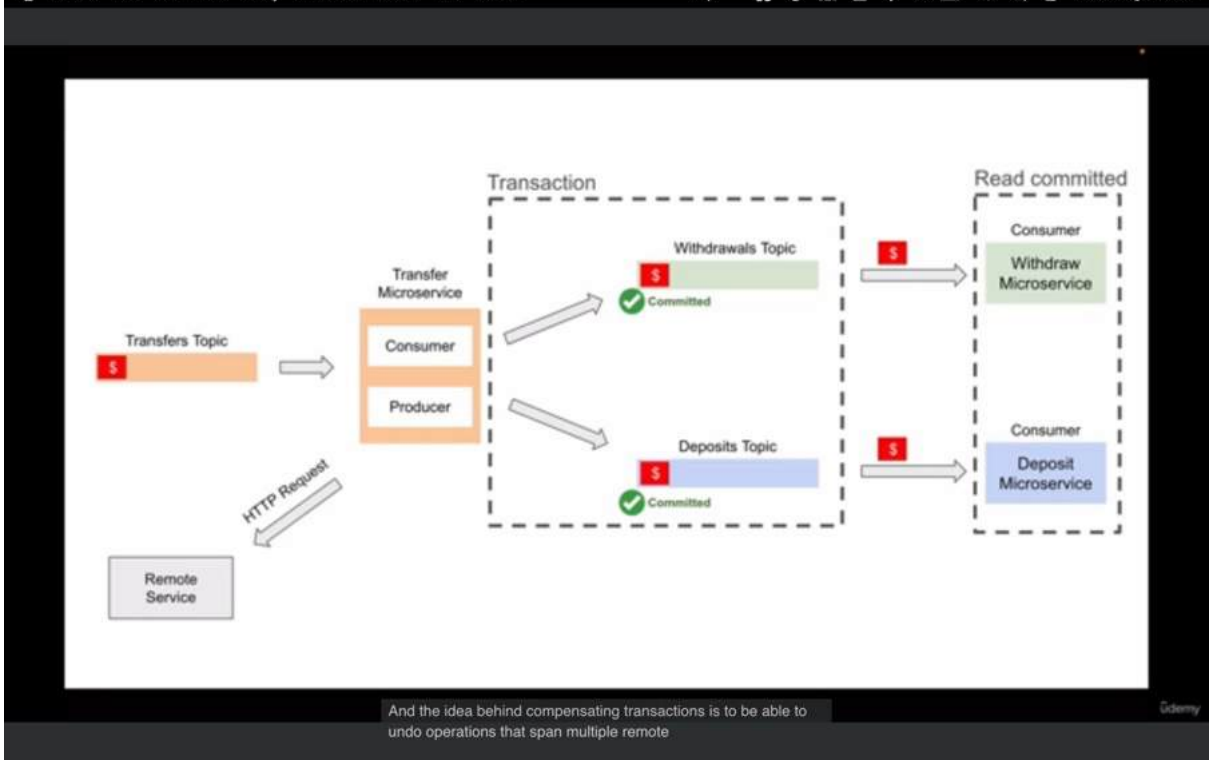

Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Uncommitted
Transaction
Read committed
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
Message in Kafka topic is marked as uncommitted and uncommitted messages will not be visible to Kafka
1.25x 3:53 / 6:40

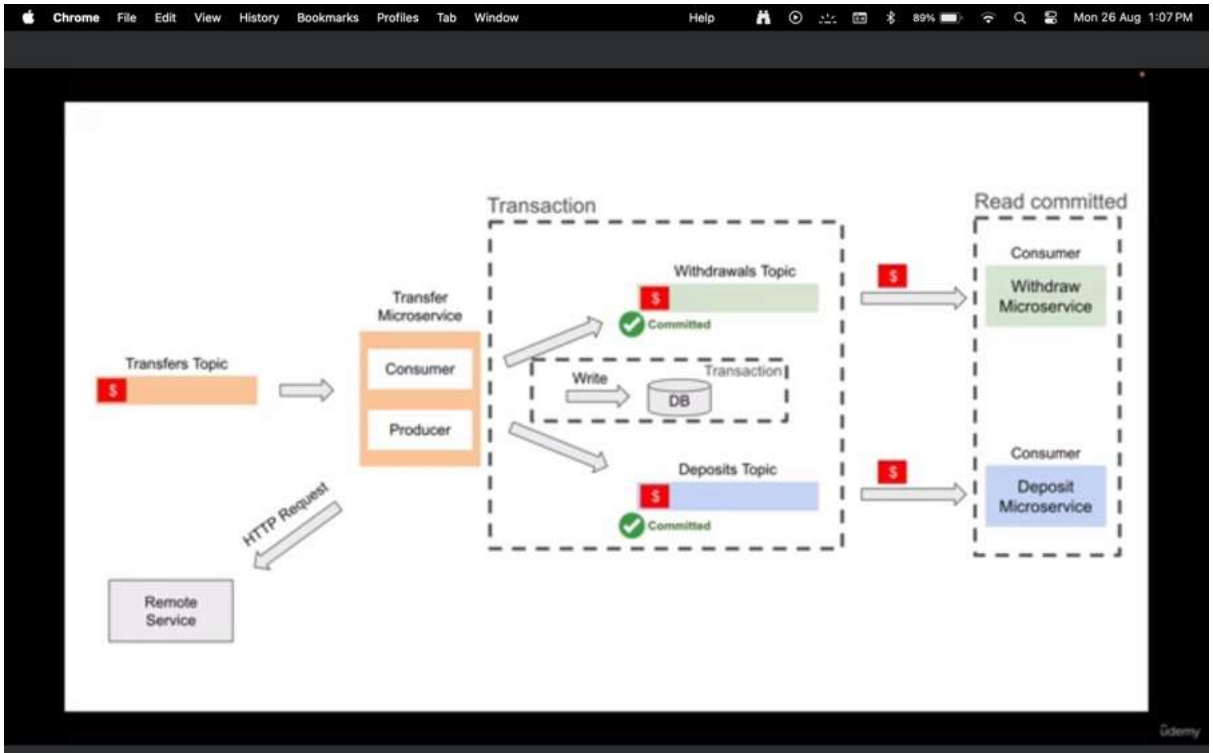

Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Uncommitted
Transaction
Read committed
Transfers Topic
Transfer Microservice
Withdrawals Topic
Deposits Topic
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
If at this moment transfer microservice crashes, or if an exception takes place, then transaction
1.25x 4:02 / 6:40


Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Aborted
Transactio
Read committed
Consumer
Withdrawals Topic
Withdraw Microservice
Transfer Microservice
Uncommitted
Transfers Topic
Consumer
Deposits Topic
Deposit Microservice
will be aborted and message will remain in topic marked as uncommitted.
1.25x 4:06 / 6:40


Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 12:24 PM
114. Introduction to Transactions in Apache Kafka
Aborted
Transactio
Read committed
Consumer
Withdrawals Topic
Withdraw Microservice
Transfer Microservice
Uncommitted
Transfers Topic
Consumer
Deposits Topic
Deposit Microservice
So technically message is still in the topic, but it is marked as uncommitted.
1.25x 4:11 / 6:40


Chrome File Edit View History Bookmarks Profiles Tab Window Help 90% Mon 26 Aug 12:25 PM
114. Introduction to Transactions in Apache Kafka
Transaction
Read committed
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Committed
Deposits Topic
Committed
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
1.25x 4:59 / 6:40


Chrome File Edit View History Bookmarks Profiles Tab Window Help 89% Mon 26 Aug 1:06 PM
Transaction
Read committed
Transfers Topic
Transfer Microservice
Consumer
Producer
Withdrawals Topic
Committed
Deposits Topic
Committed
Consumer
Withdraw Microservice
Consumer
Deposit Microservice
HTTP Request
Remote Service
And the idea behind compensating transactions is to be able to undo operations that span multiple remote



115. Demo Microservices and source code

```
public Map<String, Object> producerConfigs() {
    Map<String, Object> props = new HashMap<>();
    props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
    props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
    props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
    props.put(ProducerConfig.ACKS_CONFIG, acks);
    props.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
    props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
    props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

    props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
    props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

    return props;
}

@Bean
ProducerFactory<String, Object> producerFactory() {
    return new DefaultKafkaProducerFactory<>(producerConfigs());
}

@Bean
KafkaTemplate<String, Object> kafkaTemplate() {
    return new KafkaTemplate<String, Object>(producerFactory());
}

@Bean
NewTopic createWithdrawTopic() {
    return TopicBuilder.name(withdrawTopicName).partitions(3).replicas(3).build();
}

@Bean
NewTopic createDepositTopic() {
    return TopicBuilder.name(depositTopicName).partitions(3).replicas(3).build();
}
}
```

be used for Kafka messages to deposit money to user account.

115. Demo Microservices and source code

```java
public boolean transfer(TransferRestModel transferRestModel) {
    WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
            transferRestModel.getRecepientId(), transferRestModel.getAmount());
    DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
            transferRestModel.getRecepientId(), transferRestModel.getAmount());

    try {
        kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
                withdrawalEvent);
        LOGGER.info("Sent event to withdrawal topic.");

        // Business logic that causes and error
        callRemoteServce();

        kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
        LOGGER.info("Sent event to deposit topic");

    } catch (Exception ex) {
        LOGGER.error(ex.getMessage(), ex);
        throw new TransferServiceException(ex);
    }

    return true;
}

private ResponseEntity<String> callRemoteServce() throws Exception {
    String requestUrl = "http://localhost:8082/response/200";
    ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

    if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
        throw new Exception("Destination Microservice not availble");
    }

    if (response.getStatusCode().value() == HttpStatus.OK.value()) {
        LOGGER.info("Received
    }
```

So here in the Transfer Service class, you'll find a transfer method that receives a transfer details

116. Enable Kafka Transactions in application.properties

```properties
spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service
```

Now, when you're working with transactions in Kafka, you might have multiple producers.

116. Enable Kafka Transactions in application.properties

```
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-
```

1:57 / 5:24

```
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-
```

So this expression that starts with a dollar sign and then opening curly bracket random dot value and

```
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction=TRACE
logging.level.org.springframework.transaction=TRACE
```

Now this configuration property alone, it will be enough to enable Kafka transactions in your Spring

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

KafkaConfig.java

```
@Value("${spring.kafka.producer.properties.request.timeout.ms}")
private String requestTimeout;

@Value("${spring.kafka.producer.properties.enable.idempotence}")
private boolean idempotence;

@Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
private int inflightRequests;

public Map<String, Object> producerConfigs() {
    Map<String, Object> props = new HashMap<>();
    props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
    props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
    props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
    props.put(ProducerConfig.ACKS_CONFIG, acks);
    props.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
    props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
    props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

    props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
    props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

    return props;
}

@Bean
ProducerFactory<String, Object> producerFactory() {
    return new DefaultKafkaProducerFactory<>(producerConfigs());
}

@Bean
KafkaTemplate<String, Object> kafkaTemplate() {
    return new KafkaTemplate<String, Object>(producerFactory());
}

@Bean
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help 78% Mon 26 Aug 1:55 PM

SpringToolSuite4 File Edit Source Refactor Navigate Search Project Run Window Help Thu Mar 14 12:24 PM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

*KafkaConfig.java × application.properties

```
42
43    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
44    private String requestTimeout;
45
46    @Value("${spring.kafka.producer.properties.enable.idempotence}")
47    private boolean idempotence;
48
49    @Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
50    private int inflightRequests;
51
52    @Value("${spring.kafka.producer.transaction-id-prefix}")
53    private String transactionalIdPrefix;
54
55    public Map<String, Object> producerConfigs() {
56        Map<String, Object> props = new HashMap<>();
57        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
58        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
59        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
60        props.put(ProducerConfig.ACKS_CONFIG, acks);
61        props.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
62        props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
63        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
64
65        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
66        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);
67
68        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, "");
69
70        return props;
71    }
72
73    @Bean
74    ProducerFactory<String, Object> producerFactory() {
75        return new DefaultKafkaProducerFactory<>(producerConfigs());
76    }
77
78    @Bean
```

Writable Smart Insert 52 : 61 : 1659 Udemy

bracket.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 77% Mon 26 Aug 1:57 PM

117. Enable Kafka Transactions in the @Bean method

Thu Mar 14 12:27 PM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

*KafkaConfig.java × application.properties

```
42
43    @Value("${spring.kafka.producer.properties.request.timeout.ms}")
44    private String requestTimeout;
45
46    @Value("${spring.kafka.producer.properties.enable.idempotence}")
47    private boolean idempotence;
48
49    @Value("${spring.kafka.producer.properties.max.in.flight.requests.per.connection}")
50    private int inflightRequests;
51
52    @Value("${spring.kafka.producer.transaction-id-prefix}")
53    private String transactionalIdPrefix;
54
55    public Map<String, Object> producerConfigs() {
56        Map<String, Object> props = new HashMap<>();
57        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
58        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
59        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
60        props.put(ProducerConfig.ACKS_CONFIG, acks);
61        props.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
62        props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
63        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);
64
65        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
66        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);
67
68        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, transactionalIdPrefix);
69
70        return props;
71    }
72
73    @Bean
74    ProducerFactory<String, Object> producerFactory() {
75        return new DefaultKafkaProducerFactory<>(producerConfigs());
76    }
77
78    @Bean
```

So by setting this configuration I assign unique transaction ID to Kafka Producer.

Pause

1.25x 3:12 / 5:31

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

KafkaConfig.java × application.properties

```java
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServers);
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
        props.put(ProducerConfig.ACKS_CONFIG, acks);
        props.put(ProducerConfig.DELIVERY_TIMEOUT_MS_CONFIG, deliveryTimeout);
        props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, transactionalIdPrefix);

        return props;
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<String, Object>(producerFactory);
    }

    @Bean
    KafkaTransactionManager<String, Object> kafkaTransactionManager(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTransactionManager<>(producerFactory);
    }

    @Bean
    NewTopic createWithdrawTopic() {
        return TopicBuilder.name(withdrawTopicName).partitions(3).replicas(3).build();
    }

    @Bean
    NewTopic createDepositTopic() {
```

This should enable Kafka transactions for this microservice.

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/ui/TransfersController.java - Spring Tool Suite 4

Package Explorer ×

- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
  - src/main/java
    - com.appsdeveloperblog.estore.transfers
    - com.appsdeveloperblog.estore.transfers.error
    - com.appsdeveloperblog.estore.transfers.model
    - com.appsdeveloperblog.estore.transfers.service
    - com.appsdeveloperblog.estore.transfers.ui
      - TransfersController.java
  - src/main/resources
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
- WithdrawalService [boot]

Boot Dashboard ×

TransfersController.java ×

```java
package com.appsdeveloperblog.estore.transfers.ui;

import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/transfers")
public class TransfersController {
    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private TransferService transferService;

    public TransfersController(TransferService transferService) {
        this.transferService = transferService;
    }

    @PostMapping()
    public boolean transfer(@RequestBody TransferRestModel transferRestModel) {
        return transferService.transfer(transferRestModel);
    }
}
```

To process this information, I can open implementation of the transfer method by moving mouse pointer

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

```java
        this.environment = environment;
        this.restTemplate = restTemplate;
    }

    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferR
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-mo
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GE

        if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
            throw new Exception("Destination Microservice not availble");
        }
```

The implementation of this transfer method is in service class, which is inside of the service package.

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

```java
        this.kafkaTemplate = kafkaTemplate;
        this.environment = environment;
        this.restTemplate = restTemplate;
    }

    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferR
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-mo
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GE

        if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
            throw new Exception("Destination Microservice not availble");
```

And here's my transfer method that uses Kafka template to produce two Kafka messages.

118. Apache Kafka and @Transactional annotation

```
        RestTemplate restTemplate) {
        this.kafkaTemplate = kafkaTemplate;
        this.environment = environment;
        this.restTemplate = restTemplate;
    }

    @Transactional
    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> resp                                                    null, String.class);

        if (response.getStatusCode(
```

you tell Spring Framework that the code in this method should be executed within transaction.

Pause 1.25x 2:09 / 4:45

118. Apache Kafka and @Transactional annotation

```
        RestTemplate restTemplate) {
        this.kafkaTemplate = kafkaTemplate;
        this.environment = environment;
        this.restTemplate = restTemplate;
    }

    @Transactional(value="kafkaTransactionManager")
    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> resp                                                    null, String.class);

        if (response.getStatusCode(
```

So when you annotate this method with transactional annotation, then when the method is called Spring

Pause 1.25x 3:48 / 4:45

Chrome File Edit View History Bookmarks Profiles Tab Window Help 74% Mon 26 Aug 2:07 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 4:14 / 4:45

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

annotation, then when the method is called Spring

Framework will start a new transaction.

It will then execute all operations in this method as a single unit of work.

And if everything goes smoothly and if there are no errors, then spring will commit transaction and

consumer microservices will receive Kafka messages.

But if there is an error, like for example, a call to a remote service can fail, then Spring framework

will rollback transaction Kafka messages that were sent.

They will not be committed.

And our consumer microservices, they will not receive these messages.

Provided that we configured those consumer microservices to read only committed messages.

What we can also do is we can tell transactional

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 74% Mon 26 Aug 2:08 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 0:24 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

By default, transactional annotation does not roll back transaction.

For any exception that takes place.

By default, it rolls back transaction for unchecked exceptions and for errors.

Unchecked exceptions are also known as runtime exceptions, so by default transactional annotation will

roll back transaction for any runtime exception, but it does not roll back transaction on checked exceptions

and checked exceptions are the ones that you need to declare and handle in your method, but it is actually

configurable.

You can configure transactional annotation to roll back transaction for specific exception using rollback

for attribute.

So instead of transactional annotation right after

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 2:08 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

Unchecked exceptions and Error

RuntimeException

1.25x 0:22 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

By default, transactional annotation does not roll back transaction.

For any exception that takes place.

By default, it rolls back transaction for unchecked exceptions and for errors.

Unchecked exceptions are also known as runtime exceptions, so by default transactional annotation will

roll back transaction for any runtime exception, but it does not roll back transaction on checked exceptions

and checked exceptions are the ones that you need to declare and handle in your method, but it is actually

configurable.

You can configure transactional annotation to roll back transaction for specific exception using rollback

for attribute.

So instead of transactional annotation right

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 2:08 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

No rollback for Checked Exceptions

1.25x 0:28 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

By default, transactional annotation does not roll back transaction.

For any exception that takes place.

By default, it rolls back transaction for unchecked exceptions and for errors.

Unchecked exceptions are also known as runtime exceptions, so by default transactional annotation will

roll back transaction for any runtime exception, but it does not roll back transaction on checked exceptions

and checked exceptions are the ones that you need to declare and handle in your method, but it is actually

configurable.

You can configure transactional annotation to roll back transaction for specific exception using rollback

for attribute.

So instead of transactional annotation right

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:08 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 0:56 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

to declare and handle in your method, but it is actually

configurable.

You can configure transactional annotation to roll back transaction for specific exception using rollback

for attribute.

So instead of transactional annotation right after value attribute, you can add comma and then add

rollback for attribute and then provide the name of the exception class.

For example, if you want to rollback for all exceptions, then you will most likely use something very

general like for example throwable.

But if you want to roll back for another specific exception, then you will provide the name of that

exception here.

Runtime exceptions are rolled back by default, so ideally you'll provide here one of the checked

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:09 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 1:12 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

for attribute.

So instead of transactional annotation right after value attribute, you can add comma and then add

rollback for attribute and then provide the name of the exception class.

For example, if you want to rollback for all exceptions, then you will most likely use something very

general like for example throwable.

But if you want to roll back for another specific exception, then you will provide the name of that

exception here.

Runtime exceptions are rolled back by default, so ideally you'll provide here one of the checked exceptions

like for example SQL exception or connection exception, or one of your own custom exceptions.

For example this microservice has a custom transfer service exception.

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:09 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 1:39 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Self Service

Transcript

ideally you'll provide here one of the checked exceptions

like for example SQL exception or connection exception, or one of your own custom exceptions.

For example this microservice has a custom transfer service exception.

And if needed, you can provide more than one exception here.

And to do that, you will surround this list with curly brackets.

It will start with an opening curly bracket, then exception class comma, and then another exception

class like for example connect exception.

Something like this.

So you can provide more than one exception here.

And in this case transaction will roll back for these two exceptions now.

But even though I've specified specific exceptions

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:09 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 1:44 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

like for example SQL exception or connection exception, or one of your own custom exceptions.

For example this microservice has a custom transfer service exception.

And if needed, you can provide more than one exception here.

And to do that, you will surround this list with curly brackets.

It will start with an opening curly bracket, then exception class comma, and then another exception

class like for example connect exception.

Something like this.

So you can provide more than one exception here.

And in this case transaction will roll back for these two exceptions now.

But even though I've specified specific exceptions here.

Transaction will still roll back for any other ur

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 2:09 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

1.25x 2:00 / 4:02

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

exception here.

And to do that, you will surround this list with curly brackets.

It will start with an opening curly bracket, then exception class comma, and then another exception

class like for example connect exception.

Something like this.

So you can provide more than one exception here.

And in this case transaction will roll back for these two exceptions now.

But even though I've specified specific exceptions here.

Transaction will still roll back for any other unchecked exception.

And this is because this rollback attribute, it does not override the default behavior, which is to

roll back on runtime exceptions and for errors, it just makes sure that it does roll back transaction

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon 26 Aug 2:10 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

noRollbackFor=

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

68%

Transcript

roll back on runtime exceptions and for errors, it just makes sure that it does roll back transaction

for this specific exceptions that you indicate here.

And if needed, you can also configure this annotation, not to rollback transaction for a specific

list of exceptions.

And to do that you will use another property that is called no rollback for.

And it also can accept more than one exception.

For example, inside of curly brackets I will provide some specific exception.

Like this.

In this case, if some specific exception takes place, then transaction will not roll back.

And this is because this attribute it tells Spring framework to continue with transaction even though

this exception took place.

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:10 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

★★★★★ 68%

Transcript

not to rollback transaction for a specific

list of exceptions.

And to do that you will use another property that is called no rollback for.

And it also can accept more than one exception.

For example, inside of curly brackets I will provide some specific exception.

Like this.

In this case, if some specific exception takes place, then transaction will not roll back.

And this is because this attribute it tells Spring framework to continue with transaction even though

this exception took place.

But if your microservice application is very simple and you don't really have any preferences, like

for example, you don't have multiple transaction managers and you don't need to manage specific exceptions

Autoscroll

Chrome File Edit View History Bookmarks Profiles Tab Window Help 73% Mon 26 Aug 2:11 PM

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/4280...

New Chrome available

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learni

Student feedback

★★★★★ 68%

Transcript

But if your microservice application is very simple and you don't really have any preferences, like

for example, you don't have multiple transaction managers and you don't need to manage specific exceptions

and you're okay with default behavior, then you can remove all these attributes and use transactional

annotation as is.

I will use it as is with its default behavior, because at this moment my microservice application does

not have any other transaction managers, and I'm okay with default behavior when it comes to when to

rollback transaction.

If my spring boot application has only one transaction manager, then Spring Framework will be able

to find it and use it as default transaction manager for this method.

Okay, so I can save changes to this class and I can

Autoscroll

120. Reading committed messages in Kafka Consumer

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092

spring.kafka.consumer.group-id=amount-widthrawal-event
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsde

spring.kafka.consumer.isolation-level=read_uncommitted
```

Use capital letters to avoid red warning.
It should be: READ_UNCOMMITTED

And if you do not set this property, or if you set it explicitly to read uncommitted, then consumer

1.25x 1:30 / 5:05

120. Reading committed messages in Kafka Consumer

*application.properties

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092

spring.kafka.consumer.group-id=amount-widthrawal-event
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsde

spring.kafka.consumer.isolation-level=read_committed
```

Use capital letters to avoid red warning.
It should be: READ_COMMITTED

If I set it to read committed, then consumer microservice will read only those messages that have been

1.25x 2:00 / 5:05

120. Reading committed messages in Kafka Consumer

```java
package com.appsdeveloperblog.estore.WithdrawalService;

import java.util.HashMap;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();

        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate),
                new FixedBackOff(5000, 3));
        errorHandler.addNotRetryabl
        errorHandler.addRetryableEx
        factory.setCommonErrorHand
```

So here's a class that configures consumer factory using Java code.

1.25x 3:01 / 5:05

120. Reading committed messages in Kafka Consumer

```java
package com.appsdeveloperblog.estore.WithdrawalService;

import java.util.HashMap;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();

        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));

        config.put(ConsumerConfig.ISOLATION_LEVEL_CONFIG, environment.getProperty("spring.kafka.consumer.isolation-level"));

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        DefaultErrorHandler errorHa coverer(kafkaTemplate),
                new FixedBackOff(50
```

Unfortunately, at the time of recording this video lesson, there is a little inconsistency with the

1.25x 4:03 / 5:05

120. Reading committed messages in Kafka Consumer

```
server.port=0
spring.kafka.consumer.bootstrap-servers=localhost:9092

spring.kafka.consumer.group-id=amount-widthrawal-event
spring.kafka.consumer.properties.spring.json.trusted.packages=com.appsdeveloperblog.*

spring.kafka.consumer.isolation-level=READ_COMMITTED
```

Now it does not complain anymore.

```
package com.appsdeveloperblog.estore.WithdrawalService;

import java.util.HashMap;

@Configuration
public class KafkaConsumerConfiguration {

    @Autowired
    Environment environment;

    @Bean
    ConsumerFactory<String, Object> consumerFactory() {
        Map<String, Object> config = new HashMap<>();

        config.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG,
                environment.getProperty("spring.kafka.consumer.bootstrap-servers"));
        config.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class);
        config.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, ErrorHandlingDeserializer.class);
        config.put(ErrorHandlingDeserializer.VALUE_DESERIALIZER_CLASS, JsonDeserializer.class);
        config.put(ConsumerConfig.GROUP_ID_CONFIG, environment.getProperty("spring.kafka.consumer.group-id"));
        config.put(JsonDeserializer.TRUSTED_PACKAGES,
                environment.getProperty("spring.kafka.consumer.properties.spring.json.trusted.packages"));

        config.put(ConsumerConfig.ISOLATION_LEVEL_CONFIG,
                environment.getProperty("spring.kafka.consumer.isolation-level", "READ_COMMITED").toLowerCase());

        return new DefaultKafkaConsumerFactory<>(config);
    }

    @Bean
    ConcurrentKafkaListenerContainerFactory<String, Object> kafkaListenerContainerFactory(
            ConsumerFactory<String, Object> consumerFactory, KafkaTemplate<String, Object> kafkaTemplate) {

        ConcurrentKafkaListenerContainerFactory<String, Object> factory = new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        DefaultErrorHandler errorHandler = new DefaultErrorHandler(new DeadLetterPublishingRecoverer(kafkaTemplate),
```

Let's save changes to this class and let's continue to the next lesson.

Course: Apache Kafka for

jpmc.udemy.com/course/apache-kafka-for-spring-boot-microservices/learn/lecture/42732...

Home | Apache Kafka for Event-Driven Spring Boot Microservices | Leave a rating | Your progress | Share

Transcript

sends a Http request to another microservice.

So I will need to start it as well.

It's called mock Service and I will start it as regular spring boot application.

It's up and running and I can see its port numbers if I expand.

Spring Boot dashboard mock service is running on port number 80 and 82.

So now that my transfer microservice is running mock services running, I can start up my consumer microservices

and I will start with deposit microservice.

I will run it as Spring Boot application.

And finally I will start up withdrawal service, which I will also run as a spring boot application.

And now that my microservice is up and running, I will open postman and I will send the GTP request

with transfer details.

Autoscroll

1.25x 1:28 / 12:25

Course content | Overview | Q&A | Notes | Announcements | Reviews | Learning

Student feedback

★★★★★ 68%

122. Trying how Kafka Transactions work

Package Explorer

- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
- WithdrawalService [boot]

Boot Dashboard

- local
  - core
  - DepositService [:52934]
  - mockservice [:8082]
  - TransferService [:52887]
  - WithdrawalService [:52952]

Problems | Javadoc | Declaration | Console

```
        ssl.truststore.password = null
        ssl.truststore.type = JKS
        value.deserializer = class org.springframework.kafka.support.serializer.ErrorHandlingDes

2024-03-16T10:08:33.570-04:00  INFO 5842 --- [           main] o.a.kafka.common.utils.AppInfoPar
2024-03-16T10:08:33.571-04:00  INFO 5842 --- [           main] o.a.kafka.common.utils.AppInfoPar
2024-03-16T10:08:33.572-04:00  INFO 5842 --- [           main] o.a.kafka.common.utils.AppInfoPar
2024-03-16T10:08:33.574-04:00  INFO 5842 --- [           main] fkaConsumerFactory$ExtendedKafkaC
2024-03-16T10:08:33.586-04:00  INFO 5842 --- [           main] c.a.e.W.WithdrawalServiceApplicat
2024-03-16T10:08:33.817-04:00  INFO 5842 --- [ntainer#0-0-C-1] org.apache.kafka.clients.Metadata
2024-03-16T10:08:33.819-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:33.823-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:33.837-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:33.838-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:33.838-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.843-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.851-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.861-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.861-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.863-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.872-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
2024-03-16T10:08:36.873-04:00  INFO 5842 --- [ntainer#0-0-C-1] o.a.k.c.c.internals.ConsumerCoord
                                                 -C-1] o.a.k.c.c.internals.ConsumerCoord
                                                 -C-1] o.s.k.l.KafkaMessageListenerConta
```

And now that my microservice is up and running, I will open postman and I will send the GTP request

1.25x 1:58 / 12:25

Chrome File Edit View History Bookmarks Profiles Tab Window Help 57% Mon 26 Aug 3:16 PM

Postman File Edit View Window Help Sat Mar 16 10:12 AM

Home Workspaces API Network Upgrade

Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%. Learn more

POST http://localhost:63191 POST http://localhost:5288: No Environment

Collections APIs Environments History

http://localhost:52887/transfers Save

POST http://localhost:52887/transfers Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "senderId":"123",
3     "recepientId":"124",
4     "amount":123
5 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 1007 ms Size: 168 B Save as example

Pretty Raw Preview Visualize JSON

```
1 true
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash

The request is successful and I can now check log messages.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 56% Mon 26 Aug 3:17 PM

SpringToolSuite4 File Edit Navigate Search Project Run Window Help Sat Mar 16 10:13 AM

payments - Spring Tool Suite 4

Package Explorer
- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
- WithdrawalService [boot]

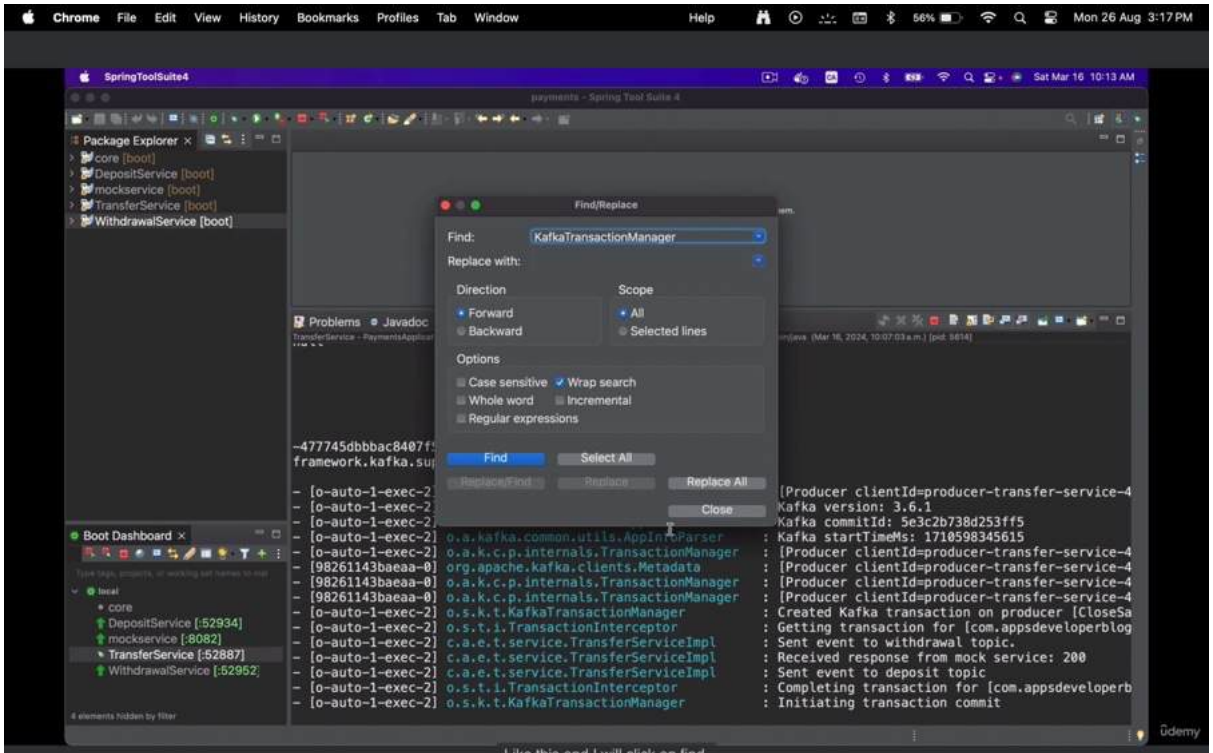
Open a file or drop files here to open them.

Problems Javadoc Declaration Console

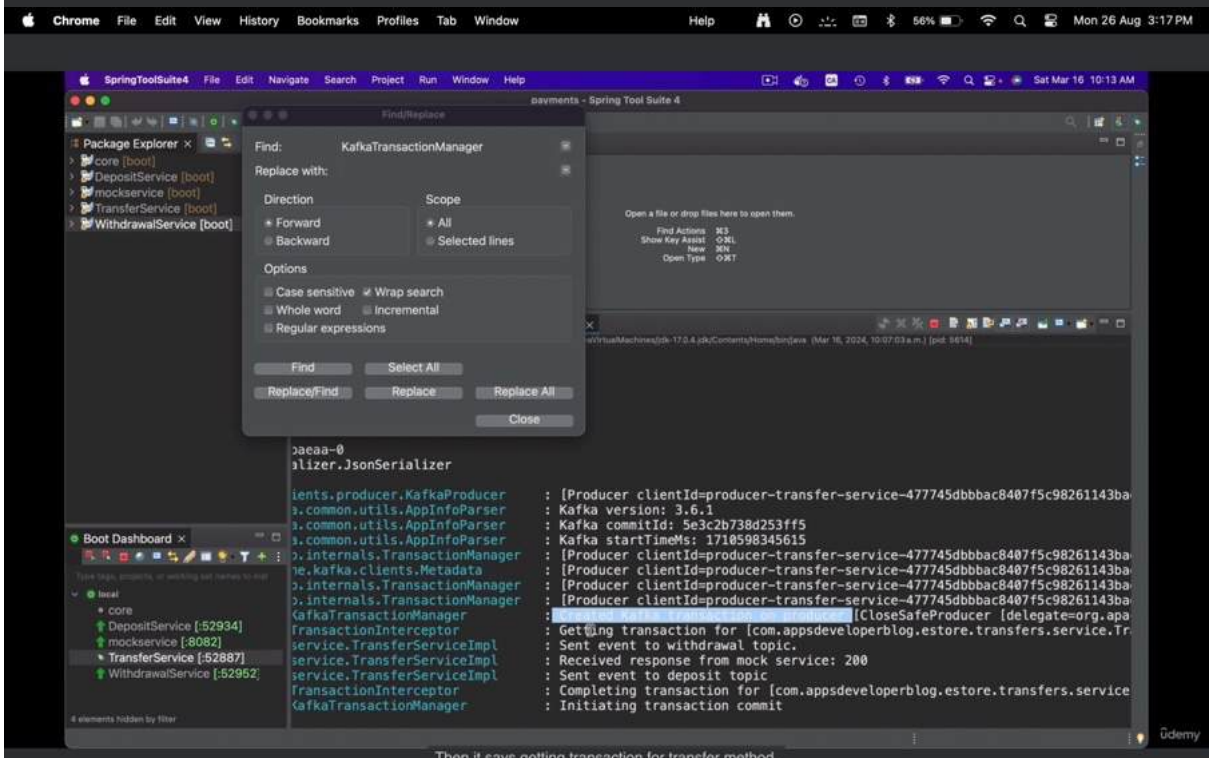
```
-477745dbbbac8407f5c98261143baeaa-0
framework.kafka.support.serializer.JsonSerializer

- [o-auto-1-exec-2] o.a.k.clients.producer.KafkaProducer     : [Producer clientId=producer-transfer-service-4
- [o-auto-1-exec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka version: 3.6.1
- [o-auto-1-exec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka commitId: 5e3c2b738d253ff5
- [o-auto-1-exec-2] o.a.kafka.common.utils.AppInfoParser     : Kafka startTimeMs: 1710598345615
- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-4
- [98261143baeaa-0] org.apache.kafka.clients.Metadata         : [Producer clientId=producer-transfer-service-4
- [98261143baeaa-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-4
- [98261143baeaa-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-4
- [o-auto-1-exec-2] o.s.k.t.KafkaTransactionManager          : Created Kafka transaction on producer [CloseSa
- [o-auto-1-exec-2] o.s.t.i.TransactionInterceptor           : Getting transaction for [com.appsdeveloperblog
- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to withdrawal topic.
- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImpl      : Received response from mock service: 200
- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to deposit topic
- [o-auto-1-exec-2] o.s.t.i.TransactionInterceptor           : Completing transaction for [com.appsdeveloperb
- [o-auto-1-exec-2] o.s.k.t.KafkaTransactionManager          : Initiating transaction commit
```

Boot Dashboard
- local
  - core
  - DepositService [:52934]
  - mockservice [:8082]
  - TransferService [:52887]
  - WithdrawalService [:52952]

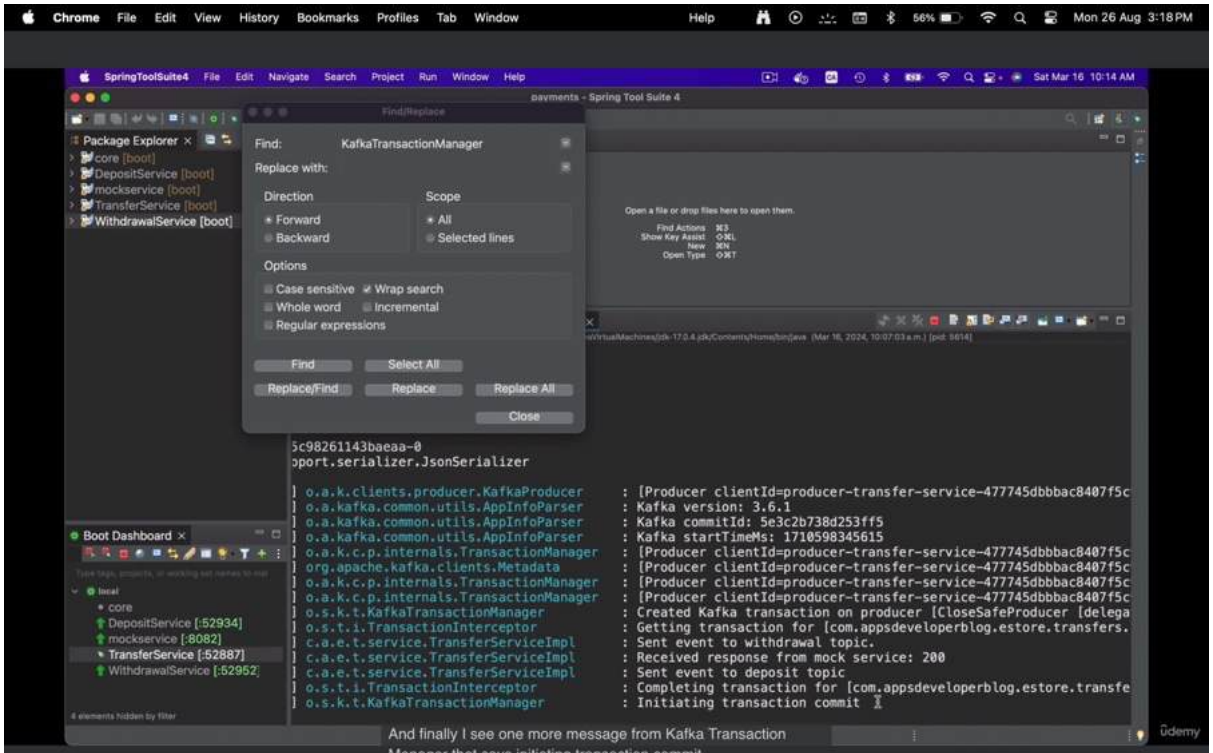
If I was able to enable Kafka transaction successfully, then I should see messages from Kafka Transaction



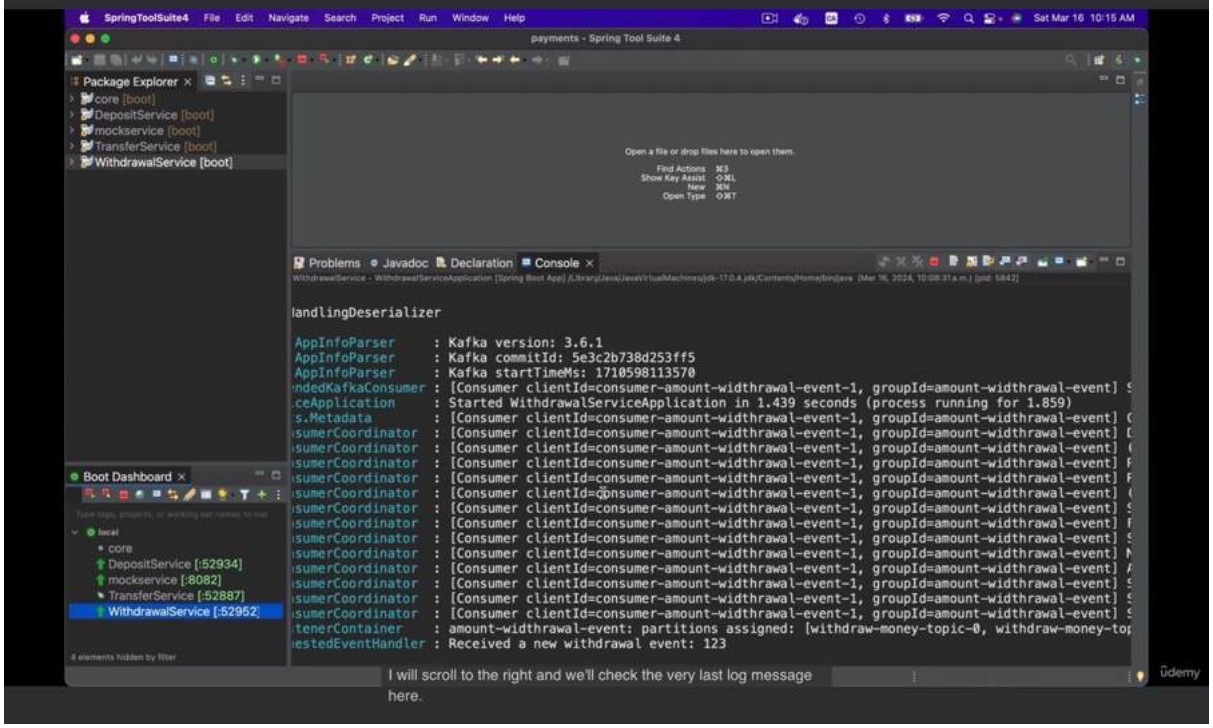
Like this and I will click on find.



Then it says getting transaction for transfer method.



And finally I see one more message from Kafka Transaction Manager that says initiating transaction commit.



I will scroll to the right and we'll check the very last log message here.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 56% Mon 26 Aug 3:19 PM

SpringToolSuite4 File Edit Navigate Search Project Run Window Help Sat Mar 16 10:16 AM

payments - Spring Tool Suite 4

Package Explorer
- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
- WithdrawalService [boot]

Boot Dashboard
- local
  - core
  - DepositService [:52934]
  - mockservice [:8082]
  - TransferService [:52887]
  - WithdrawalService [:52952]

Problems Javadoc Declaration Console

```
alizer.ErrorHandlingDeserializer

common.utils.AppInfoParser      : Kafka version: 3.6.1
common.utils.AppInfoParser      : Kafka commitId: 5e3c2b738d253ff5
common.utils.AppInfoParser      : Kafka startTimeMs: 1710598113570
rFactory$ExtendedKafkaConsumer : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
thdrawalServiceApplication      : Started WithdrawalServiceApplication in 1.439 seconds (process running for 1
.kafka.clients.Metadata         : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
internals.ConsumerCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
fkaMessageListenerContainer     : amount-widthrawal-event: partitions assigned: [withdraw-money-topic-0, withd
ithdrawalRequestedEventHandler  : Received a new withdrawal event: 123
```

So I will select mock service and we'll click on the stop button.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 56% Mon 26 Aug 3:19 PM

SpringToolSuite4 File Edit Source Refactor Navigate Search Project Run Window Help Sat Mar 16 10:17 AM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

TransfersController.java | TransferServiceImpl.java

```
31          this.kafkaTemplate = kafkaTemplate;
32          this.environment = environment;
33          this.restTemplate = restTemplate;
34      }
35
36      @Transactional
37      @Override
38      public boolean transfer(TransferRestModel transferRestModel) {
39          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequeste
40                  transferRestModel.getRecepientId(), transferRestModel.get
41          DepositRequestedEvent depositEvent = new DepositRequestedEvent(tr
42                  transferRestModel.getRecepientId(), transferRestModel.get
43
44          try {
45              kafkaTemplate.send(environment.getProperty("withdraw-money-to
46                      withdrawalEvent);
47              LOGGER.info("Sent event to withdrawal topic.");
48
49              // Business logic that causes and error
50              callRemoteServce();
51
52              kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
53              LOGGER.info("Sent event to deposit topic");
54
55          } catch (Exception ex) {
56              LOGGER.error(ex.getMessage(), ex);
57              throw new TransferServiceException(ex);
58          }
59
60          return true;
61      }
62
63      private ResponseEntity<String> callRemoteServce() throws Exception {
64          String requestUrl = "http://localhost:8082/response/200";
65          ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);
66
67          if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
```

Problems Javadoc Declaration Console

```
5c98261143baeaa-0
pport.serializer.JsonSerializer

] o.a.k.clients.producer.KafkaProducer      : [Producer cli
] o.a.kafka.common.utils.AppInfoParser      : Kafka version
] o.a.kafka.common.utils.AppInfoParser      : Kafka commitI
] o.a.kafka.common.utils.AppInfoParser      : Kafka startTi
] o.a.k.c.p.internals.TransactionManager    : [Producer cli
] org.apache.kafka.clients.Metadata         : [Producer cli
] o.a.k.c.p.internals.TransactionManager    : [Producer cli
] o.a.k.c.p.internals.TransactionManager    : [Producer cli
] o.s.k.t.KafkaTransactionManager           : Created Kafka
] o.s.t.i.TransactionInterceptor            : Getting trans
```

And then it sends another Kafka message to a deposit topic.

122. Trying how Kafka Transactions work

Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%.

POST http://localhost:52887/transfers

```
{
    "senderId":"123",
    "recepientId":"124",
    "amount":123
}
```

Status: 200 OK Time: 1007 ms Size: 168 B

```
true
```

I will go back to postman and send this request.

1.25x 9:01 / 12:25

122. Trying how Kafka Transactions work

```
        this.kafkaTemplate = kafkaTemplate;
        this.environment = environment;
        this.restTemplate = restTemplate;
    }

    @Transactional
    @Override
    public boolean transfer(TransferRestModel transferRestMode
        WithdrawalRequestedEvent withdrawalEvent = new Withdra
                transferRestModel.getRecepientId(), transferRe
        DepositRequestedEvent depositEvent = new DepositReques
                transferRestModel.getRecepientId(), transferRe

        try {
            kafkaTemplate.send(environment.getProperty("withdr
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposi
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
```

```
org.apache.kafka.clients.Metadata        : [Producer clientId=producer-transfer-service-477745dbbbac840
org.apache.kafka.clients.Metadata        : [Producer clientId=producer-transfer-service-477745dbbbac840
o.s.k.t.KafkaTransactionManager          : Creating new transaction with name [com.appsdeveloperblog.es
o.s.k.t.KafkaTransactionManager          : Created Kafka transaction on producer [CloseSafeProducer [de
o.s.                                     : for [com.appsdeveloperblog.estore.transf
c.a.                                     rawal topic.
```

And now the debugger breakpoint is on the line that is about to call remote service.

1.25x 9:26 / 12:25

122. Trying how Kafka Transactions work

```
31        this.kafkaTemplate = kafkaTemplate;
32        this.environment = environment;
33        this.restTemplate = restTemplate;
34    }
35
36    @Transactional
37    @Override
38    public boolean transfer(TransferRestModel transferRestMode
39        WithdrawalRequestedEvent withdrawalEvent = new Withdra
```

```
12:26.133-04:00  INFO 5614 --- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to
12:26.134-04:00 TRACE 5614 --- [o-auto-1-exec-2] o.s.t.i.TransactionInterceptor           : Completing tr
12:26.134-04:00 DEBUG 5614 --- [o-auto-1-exec-2] o.s.k.t.KafkaTransactionManager          : Initiating tr
17:25.642-04:00  INFO 5614 --- [98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer cli
17:25.643-04:00  INFO 5614 --- [98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer cli
17:25.643-04:00  INFO 5614 --- [98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer cli
18:54.008-04:00 DEBUG 5614 --- [o-auto-1-exec-5] o.s.k.t.KafkaTransactionManager          : Creating new
18:54.008-04:00 DEBUG 5614 --- [o-auto-1-exec-5] o.s.k.t.KafkaTransactionManager          : Created Kafka
18:54.008-04:00 TRACE 5614 --- [o-auto-1-exec-5] o.s.t.i.TransactionInterceptor           : Getting trans
19:46.177-04:00  INFO 5614 --- [o-auto-1-exec-5] c.a.e.t.service.TransferServiceImpl      : Sent event to
20:05.339-04:00 ERROR 5614 --- [o-auto-1-exec-5] c.a.e.t.service.TransferServiceImpl      : I/O error on

ework.web.client.ResourceAccessException: I/O error on GET request for "http://localhost:8082/response/20
.springframework.web.client.RestTemplate.createResourceAccessException(RestTemplate.java:915) ~[spring-we
.springframework.web.client.RestTemplate.doExecute(RestTemplate.java:895) ~[spring-web-6.1.4.jar:6.1.4]
.springframework.web.client.RestTemplate.execute(RestTemplate.java:790) ~[spring-web-6.1.4.jar:6.1.4]
.springframework.web.client.RestTemplate.exchange(RestTemplate.java:672) ~[spring-web-6.1.4.jar:6.1.4]
.appsdeveloperblog.estore.transfers.service.TransferServiceImpl.callRemoteServce(TransferServiceImpl.java
.appsdeveloperblog.estore.transfers.service.TransferServiceImpl.transfer(TransferServiceImpl.java:50) ~[c
a.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) ~[na:na]
a.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) ~[na:na]
a.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[n
a.base/java.lang.reflect.Method.invoke(Method.java:568) ~[na:na]
.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:351) ~[spring-aop-6.1.
.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(ReflectiveMethodInvocation.java
```

1.25x 9:53 / 12:25

122. Trying how Kafka Transactions work

```
31        this.kafkaTemplate = kafkaTemplate;
32        this.environment = environment;
33        this.restTemplate = restTemplate;
34    }
35
36    @Transactional
37    @Override
38    public boolean transfer(TransferRestModel transferRestMode
39        WithdrawalRequestedEvent withdrawalEvent = new Withdra
```

```
[o-auto-1-exec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to deposit topic
[o-auto-1-exec-2] o.s.t.i.TransactionInterceptor           : Completing transaction for [com.appsdevelope
[o-auto-1-exec-2] o.s.k.t.KafkaTransactionManager          : Initiating transaction commit
[98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer clientId=producer-transfer-service
[98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer clientId=producer-transfer-service
[98261143baeaa-0] org.apache.kafka.clients.Metadata        : [Producer clientId=producer-transfer-service
[o-auto-1-exec-5] o.s.k.t.KafkaTransactionManager          : Creating new transaction with name [com.apps
[o-auto-1-exec-5] o.s.k.t.KafkaTransactionManager          : Created Kafka transaction on producer [Close
[o-auto-1-exec-5] o.s.t.i.TransactionInterceptor           : Getting transaction for [com.appsdeveloperbl
[o-auto-1-exec-5] c.a.e.t.service.TransferServiceImpl      : Sent event to withdrawal topic.
[o-auto-1-exec-5] c.a.e.t.service.TransferServiceImpl      : I/O error on GET request for "http://localho

Exception: I/O error on GET request for "http://localhost:8082/response/200": Connection refused
tTemplate.createResourceAccessException(RestTemplate.java:915) ~[spring-web-6.1.4.jar:6.1.4]
tTemplate.doExecute(RestTemplate.java:895) ~[spring-web-6.1.4.jar:6.1.4]
tTemplate.execute(RestTemplate.java:790) ~[spring-web-6.1.4.jar:6.1.4]
tTemplate.exchange(RestTemplate.java:672) ~[spring-web-6.1.4.jar:6.1.4]
fers.service.TransferServiceImpl.callRemoteServce(TransferServiceImpl.java:65) ~[classes/:na]
fers.service.TransferServiceImpl.transfer(TransferServiceImpl.java:50) ~[classes/:na]
iveMethodAccessorImpl.invoke0(Native Method) ~[na:na]
iveMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) ~[na:na]
egatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) ~[na:na]
.invoke(Method.java:568) ~[na:na]
pUtils.invokeJoinpointUsingReflection(AopUtils.java:351) ~[spring-aop-6.1.4.jar:6.1.4]
ReflectiveMethodInvocatio                                  eflectiveMethodInvocation.java:196) ~[spring-aop-6.
```

Connection refused.

1.25x 9:55 / 12:25

122. Trying how Kafka Transactions work

```
uto-1-exec-5] o.s.t.i.TransactionInterceptor          : Completing transaction for [com.appsdeveloperblo
uto-1-exec-5] o.s.k.t.KafkaTransactionManager         : Initiating transaction rollback
uto-1-exec-5] o.a.k.clients.producer.KafkaProducer    : [Producer clientId=producer-transfer-service-477
uto-1-exec-5] o.a.c.c.C.[.[.[/].[dispatcherServlet]   : Servlet.service() for servlet [dispatcherServlet
```

1.25x 11:05 / 12:25

122. Trying how Kafka Transactions work

```
uto-1-exec-5] o.s.t.i.TransactionInterceptor          : Completing transaction for [com.appsdeveloperblo
uto-1-exec-5] o.s.k.t.KafkaTransactionManager         : Initiating transaction rollback
uto-1-exec-5] o.a.k.clients.producer.KafkaProducer    : [Producer clientId=producer-transfer-service-477
uto-1-exec-5] o.a.c.c.C.[.[.[/].[dispatcherServlet]   : Servlet.service() for servlet [dispatcherServlet
```

1.25x 11:23 / 12:25

Chrome File Edit View History Bookmarks Profiles Tab Window Help 55% Mon 26 Aug 3:22 PM

Postman File Edit View Window Help Sat Mar 16 10:26 AM

Home Workspaces API Network Upgrade

Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%. Learn more

POST http://localhost:63191 POST http://localhost:5288: No Environment

http://localhost:52887/transfers Save

POST http://localhost:52887/transfers Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2     "senderId":"123",
3     "recepientId":"124",
4     "amount":123
5 }
```

Body Cookies Headers (4) Test Results Status: 500 Internal Server Error Time: 1 m 11.46 s Size: 264 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2     "timestamp": "2024-03-16T14:20:05.413+00:00",
3     "status": 500,
4     "error": "Internal Server Error",
5     "path": "/transfers"
6 }
```

Online Find and replace Console Runner Start Proxy Cookies Trash Udemy

I will send a second message, but with a different amount so that we see it better.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 55% Mon 26 Aug 3:22 PM

122. Trying how Kafka Transactions work

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

Debug Project Ex... Servers

TransferService - PaymentsApplication [Spr
com.appsdeveloperblog.estore.transfers.F
Thread [Catalina-utility-1] (Running)
Thread [Catalina-utility-2] (Running)
Thread [container-0] (Running)
Daemon Thread [http-nio-auto-1-exec-1
Daemon Thread [http-nio-auto-1-exec-2
Daemon Thread [http-nio-auto-1-exec-3
Daemon Thread [http-nio-auto-1-exec-4
Daemon Thread [http-nio-auto-1-exec-5
Daemon Thread [http-nio-auto-1-exec-6
Daemon Thread [http-nio-auto-1-exec-7
Daemon Thread [http-nio-auto-1-exec-8
Daemon Thread [http-nio-auto-1-exec-9
Daemon Thread [http-nio-auto-1-exec-10
Daemon Thread [http-nio-auto-1-Poller]
Daemon Thread [http-nio-auto-1-Accep
Thread [DestroyJavaVM] (Running)
Daemon Thread [kafka-producer-netwo
/Library/Java/JavaVirtualMachines/jdk-17.0
<terminated>mockservice - MockserviceAp
<terminated, exit value: 0>/Library/Java/Ja
DepositService - DepositServiceApplication
/Library/Java/JavaVirtualMachines/jdk-17.0
WithdrawalService - WithdrawalServiceAppl
/Library/Java/JavaVirtualMachines/jdk-17.0

TransfersController.java TransferServiceImpl.java

```
31          this.kafkaTemplate = kafkaTemplate;
32          this.environment = environment;
33          this.restTemplate = restTemplate;
34      }
35
36      @Transactional
37      @Override
38      public boolean transfer(TransferRestModel transferRestMode
39          WithdrawalRequestedEvent withdrawalEvent = new Withdra
```

Variables Breakp... Express...

Console Problems Debug Shell

```
igDeserializer

oParser      : Kafka version: 3.6.1
oParser      : Kafka commitId: 5e3c2b738d253ff5
oParser      : Kafka startTimeMs: 1710598113570
fkaConsumer  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Sul
ication      : Started WithdrawalServiceApplication in 1.439 seconds (process running for 1.859)
data         : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Cl
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Di
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] (R
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Re
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Re
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] (R
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Su
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Fi
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Su
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] No
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event] Ad
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
oordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthr
ontainer     : amount-widthrawal-event: partitions assigned: [withdraw-money-topic-0, withd
ventHandler  : Received a new withdrawal event: 123
orkClient    : [Consumer                                    unt-widthrawal-event-1, groupId=amount-widthr
```

Now let's send it again.

1.25x 11:31 / 12:25

```
try {
    kafkaTemplate.send(environment.getProperty("withdr
            withdrawalEvent);
    LOGGER.info("Sent event to withdrawal topic.");

    // Business logic that causes and error
    callRemoteServce();

    kafkaTemplate.send(environment.getProperty("deposi
```

```
lingDeserializer
InfoParser      : Kafka version: 3.6.1
InfoParser      : Kafka commitId: 5e3c2b738d253ff5
InfoParser      : Kafka startTimeMs: 1710598113570
dKafkaConsumer  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
pplication      : Started WithdrawalServiceApplication in 1.439 seconds (process running for 1.859)
etadata         : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erCoordinator   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
erContainer     : amount-widthrawal-event: partitions assigned: [withdraw-money-topic-0, withdraw-money-t
edEventHandler  : Received a new withdrawal event: 123
etworkClient    : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawal-event]
```

There are no messages with the amounts 1 to 4 or 1 to 5.

123. Apache Kafka local transactions with KafkaTemplate

```
boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

    t.send("withdraw-money-topic", withdrawalEvent);

    t.send("deposit-money-topic", depositEvent);

        return true;

});
```

For example, you can use it to send two Kafka messages and the t object here that is being used to

1.25x 1:05 / 5:11

123. Apache Kafka local transactions with KafkaTemplate

```
boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

        t.send("withdraw-money-topic", withdrawalEvent);

          throws Exception
        callRemoteServce();

        t.send("deposit-money-topic", depositEvent);

        return true;

});
```

service and this call results in exception thrown.

1.25x 1:26 / 5:11

123. Apache Kafka local transactions with KafkaTemplate

Scope of the Transaction

```
public void transfer(TransferRestModel transferRestModel) {

        boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

            t.send("withdraw-money-topic", withdrawalEvent);

            t.send("deposit-money-topic", depositEvent);

            return true;
        });

        if(true) throw new Exception("Forcing exception");

}
```

And this is because this exception did not take place within local transaction.

Pause

1.25x 2:44 / 5:11

Scope of the Transaction

Managed by Spring framework

Managed by Kafka's internal Transaction manager

```
@Transactional("kafkaTransactionManager")
public void transfer(TransferRestModel transferRestModel) {

    boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

        t.send("withdraw-money-topic", withdrawalEvent);

        t.send("deposit-money-topic", depositEvent);

        return true;
    });

    if(true) throw new Exception("Forcing exception");

}
```

It is managed by Kafka's internal transactional manager component and to lock activity of Kafka's internal

Scope of the Transaction

Managed by Spring framework

Managed by Kafka's internal Transaction manager

```
@Transactional("kafkaTransactionManager")
public void transfer(TransferRestModel transferRestModel) {

    boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

        t.send("withdraw-money-topic", withdrawalEvent);

        t.send("deposit-money-topic", depositEvent);

        return true;
    });

    if(true) throw new Exception("Forcing exception");

}
```

logging.level.org.apache.kafka.clients.producer.internals.TransactionManager=TRACE

will configure login level for transaction Manager that is provided by Apache Kafka.

Scope of the Transaction

Managed by Spring framework

Managed by Kafka's internal Transaction manager

```
@Transactional("kafkaTransactionManager")
public void transfer(TransferRestModel transferRestModel) {

    boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

        t.send("withdraw-money-topic", withdrawalEvent);

        t.send("deposit-money-topic", depositEvent);

        return true;
    });

    if(true) throw new Exception("Forcing exception");

}
```

logging.level.org.apache.kafka.clients.producer.internals.TransactionManager=TRACE

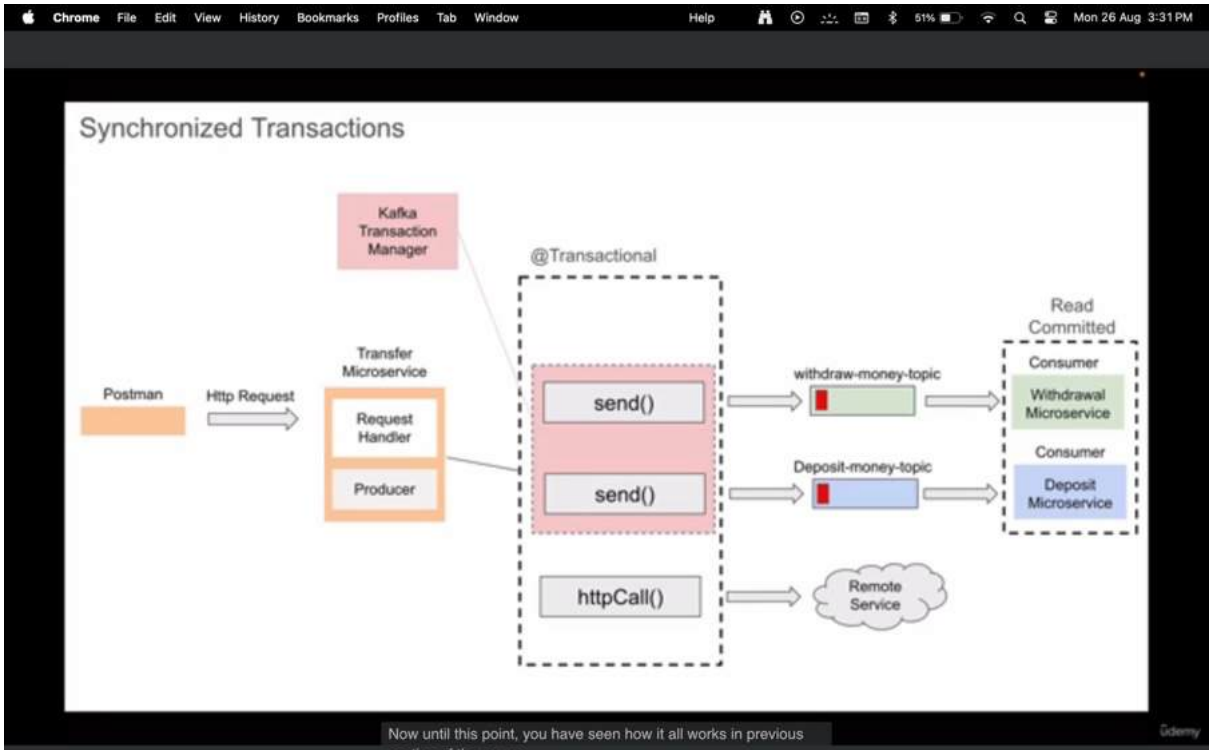
logging.level.org.springframework.kafka.transaction.KafkaTransactionManager=TRACE

management infrastructure, and this integration allows Kafka transactions to participate in Springs

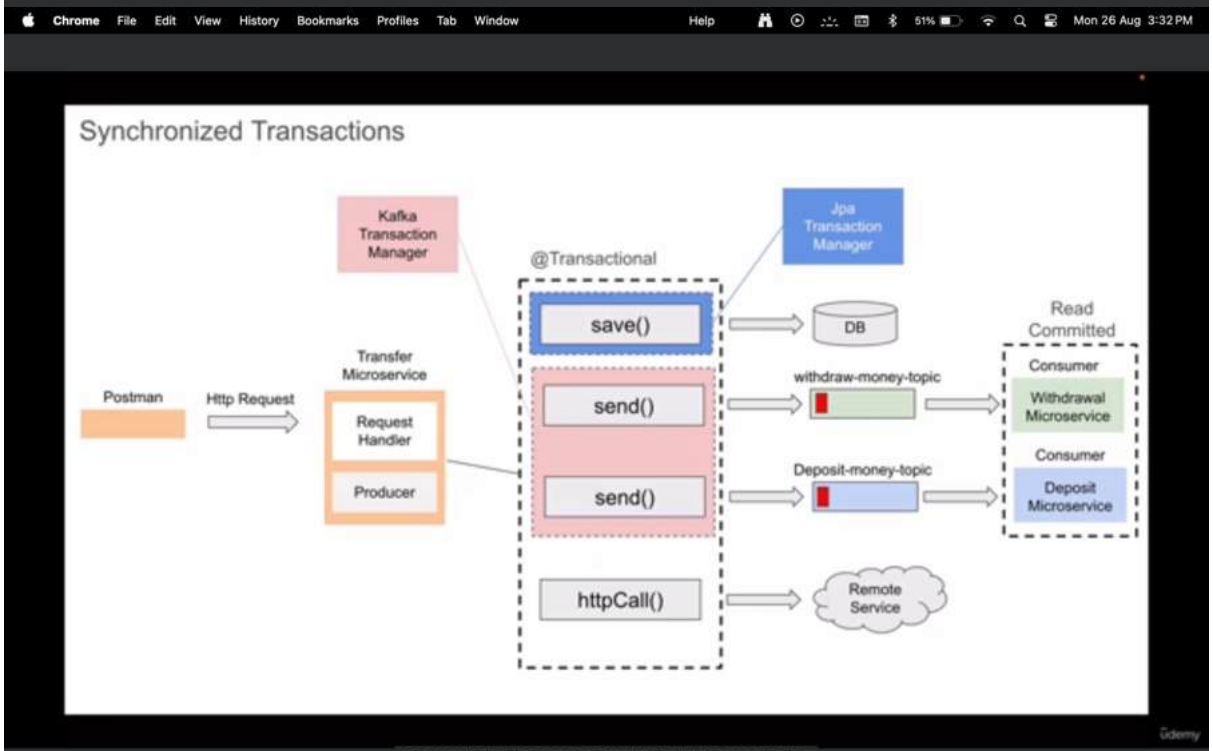
Scope of the Transaction

```
@Transactional("kafkaTransactionManager")
public void transfer(TransferRestModel transferRestModel) {

    boolean returnValue = kafkaTemplate.executeInTransaction(t -> {

        t.send("withdraw-money-topic", withdrawalEvent);

        t.send("deposit-money-topic", depositEvent);

        return true;
    });

}
```

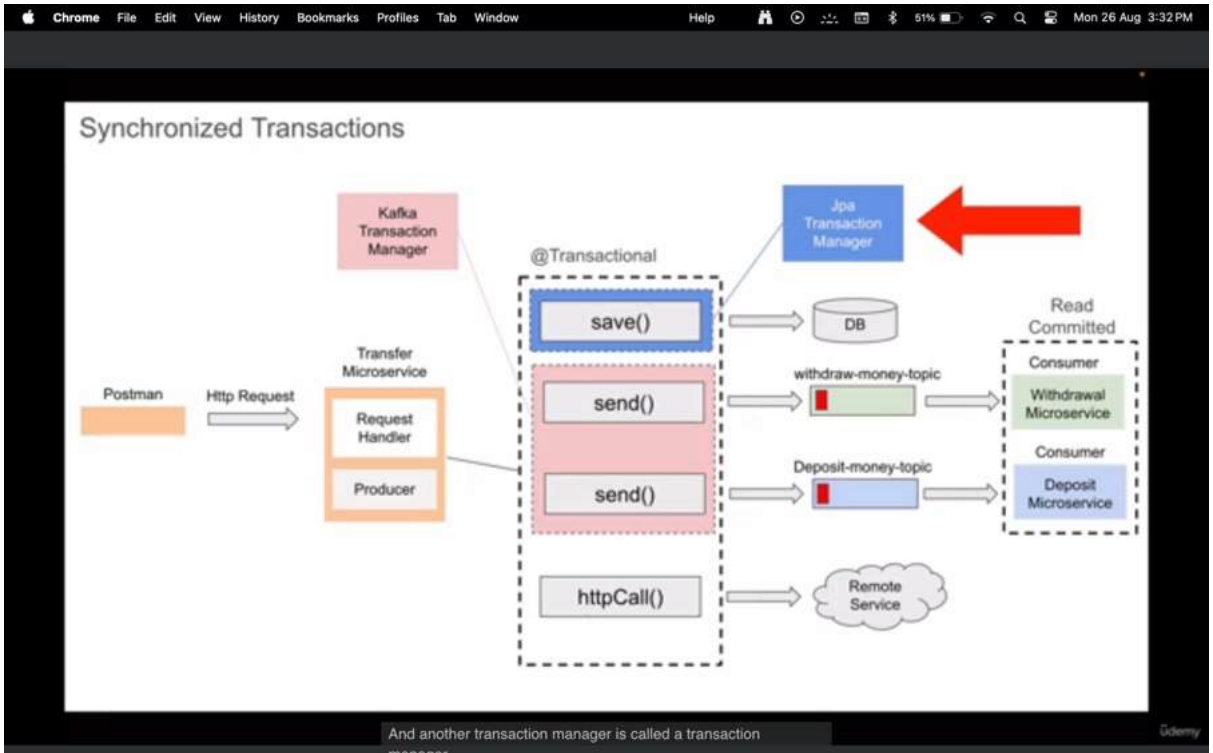
annotation, because otherwise you're creating nested transaction you're starting with.

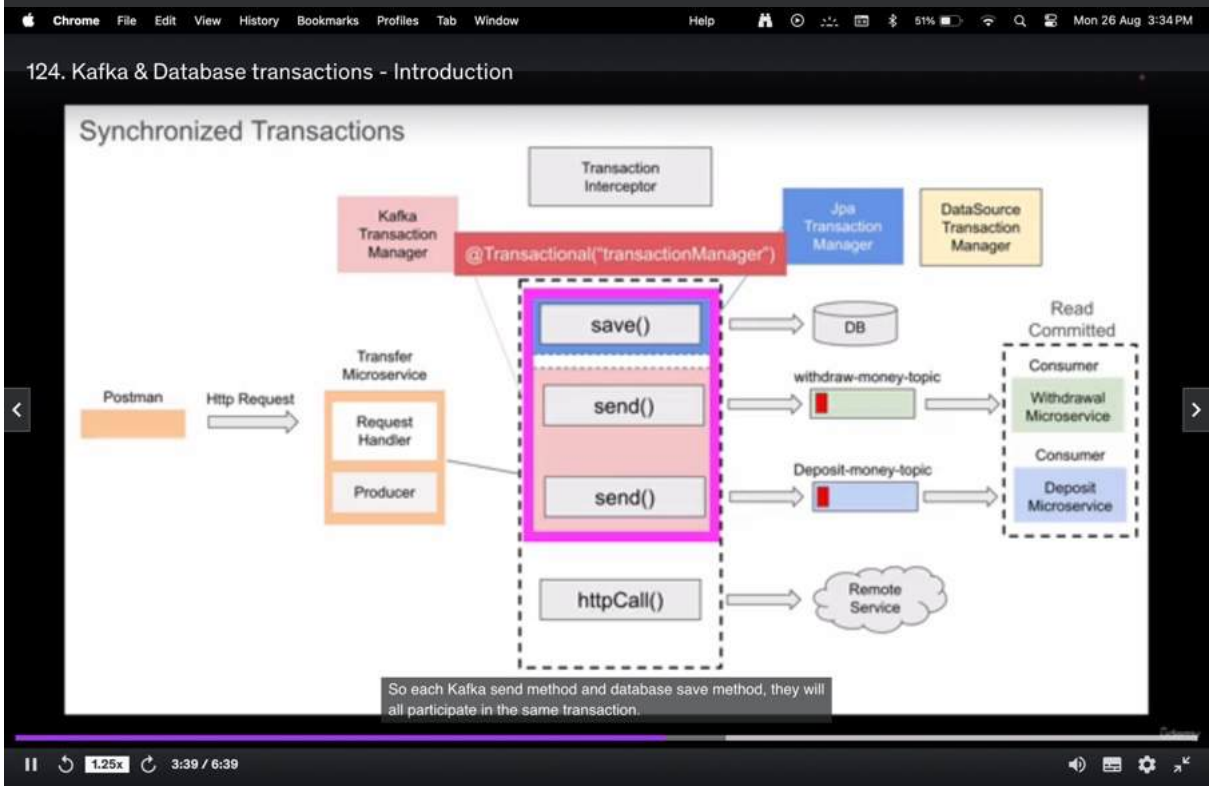


Now until this point, you have seen how it all works in previous section of the course.

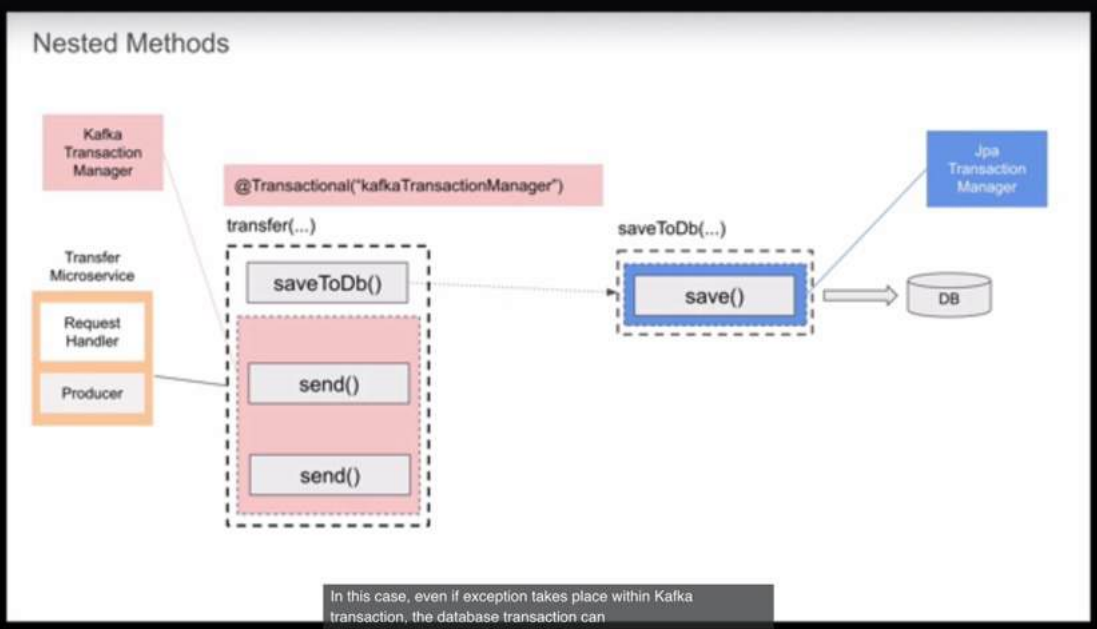


manager that will manage transaction for database operation.

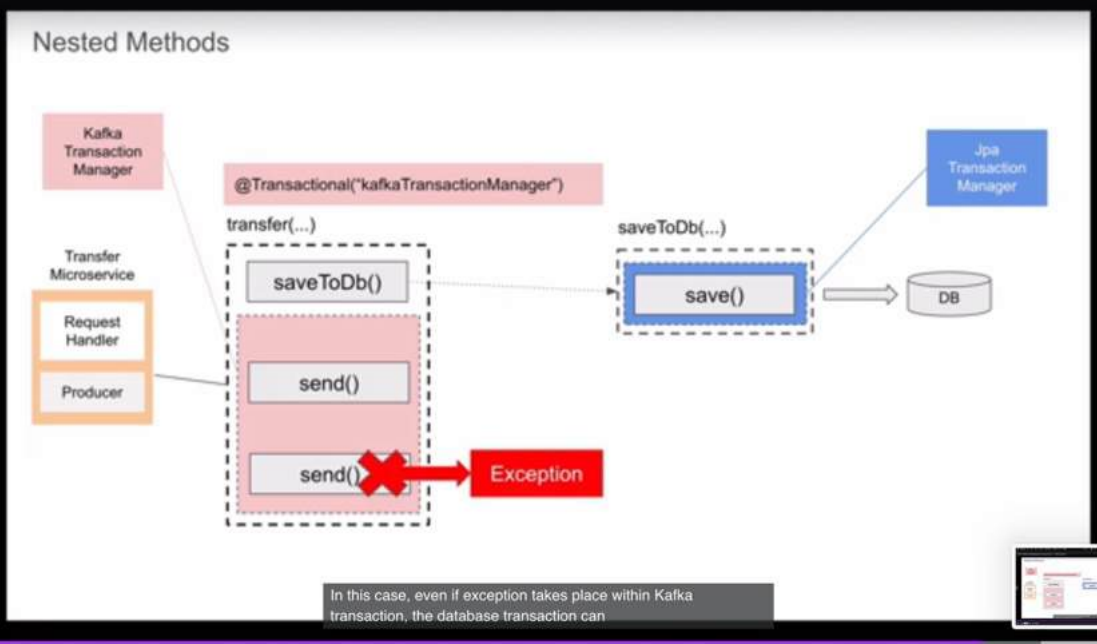

Chrome File Edit View History Bookmarks Profiles Tab Window Help 51% Mon 26 Aug 3:32 PM
Synchronized Transactions
Kafka Transaction Manager
@Transactional
Jpa Transaction Manager
save()
DB
Read Committed
Transfer Microservice
Postman
Http Request
Request Handler
Producer
send()
withdraw-money-topic
Consumer
Withdrawal Microservice
send()
Deposit-money-topic
Consumer
Deposit Microservice
httpCall()
Remote Service
And another transaction manager is called a transaction manager.


Chrome File Edit View History Bookmarks Profiles Tab Window Help 51% Mon 26 Aug 3:34 PM
124. Kafka & Database transactions - Introduction
Synchronized Transactions
Transaction Interceptor
Kafka Transaction Manager
@Transactional("transactionManager")
Jpa Transaction Manager
DataSource Transaction Manager
save()
DB
Read Committed
Transfer Microservice
Postman
Http Request
Request Handler
Producer
send()
withdraw-money-topic
Consumer
Withdrawal Microservice
send()
Deposit-money-topic
Consumer
Deposit Microservice
httpCall()
Remote Service
So each Kafka send method and database save method, they will all participate in the same transaction.
1.25x
3:39 / 6:39

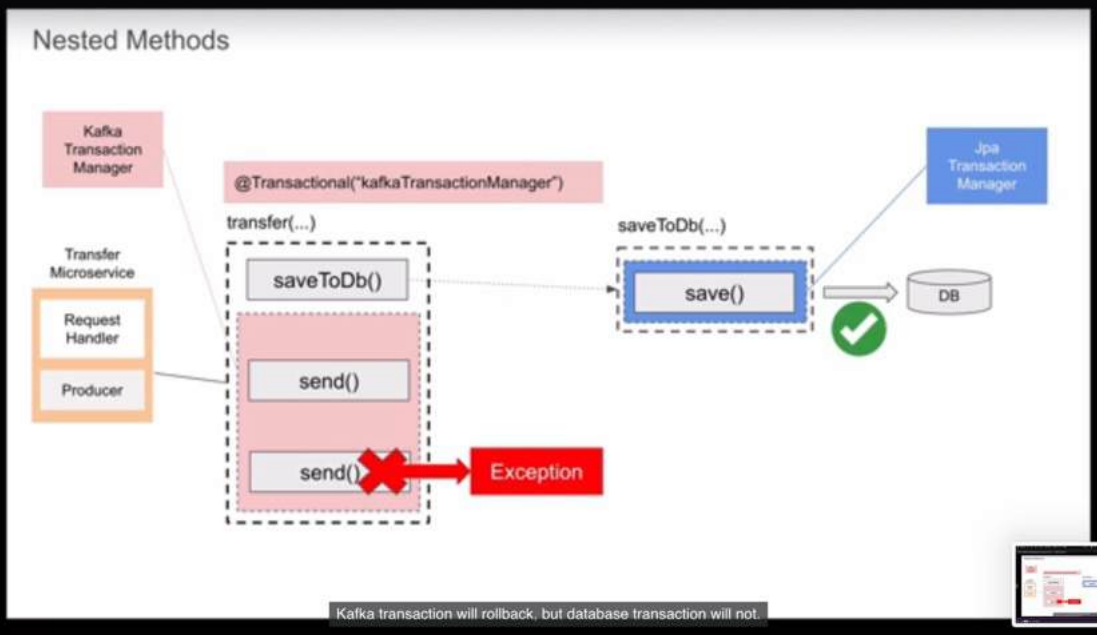
124. Kafka & Database transactions - Introduction



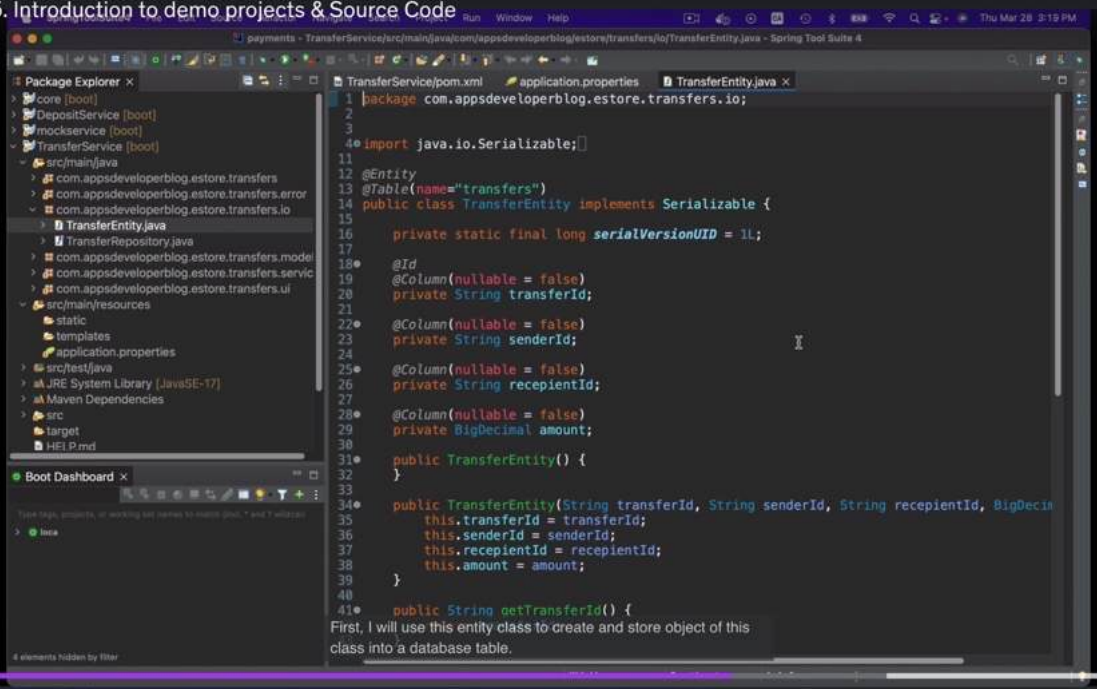
124. Kafka & Database transactions - Introduction



124. Kafka & Database transactions - Introduction



125. Introduction to demo projects & Source Code



126. Creating Jpa Transaction Manager

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/ui/TransfersController.java - Spring Tool Suite 4

```java
package com.appsdeveloperblog.estore.transfers.ui;

import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/transfers")
public class TransfersController {
    private final Logger LOGGER = LoggerFactory.getLogger(this.getClass());
    private TransferService transferService;

    public TransfersController(TransferService transferService) {
        this.transferService = transferService;
    }

    @PostMapping()
    public boolean transfer(@RequestBody TransferRestModel transferRestModel) {
        return transferService.transfer(transferRestModel);
    }
}
```

If I open it then here I have a method that handles http post request.

1.25x 0:24 / 5:19

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

```java
        this.restTemplate = restTemplate;
    }

    @Transactional
    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

        if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
            throw new Exception("Destination Microservice not availble");
        }
```

And here in the transfer method, I sent two messages to two different Kafka topics, and to send this

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

TransfersController.java | TransferServiceImpl.java | KafkaConfig.java

```java
        props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, transactionalIdPrefix);

        return props;
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean
    KafkaTransactionManager<String, Object> kafkaTransactionManager(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTransactionManager<>(producerFactory);
    }

    @Bean
    NewTopic createWithdrawTopic() {
        return TopicBuilder.name(withdrawTopicName).partitions(3).replicas(3).build();
    }

    @Bean
    NewTopic createDepositTopic() {
        return TopicBuilder.name(depositTopicName).partitions(3).replicas(3).build();
    }
}
```

And here I have a method that creates a new Kafka transactional manager, and we did it in previous

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

TransfersController.java | TransferServiceImpl.java | *KafkaConfig.java

```java
        props.put(ProducerConfig.LINGER_MS_CONFIG, linger);
        props.put(ProducerConfig.REQUEST_TIMEOUT_MS_CONFIG, requestTimeout);

        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, transactionalIdPrefix);

        return props;
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean("kafkaTransactionManager")
    KafkaTransactionManager<String, Object> kafkaTransactionManager(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTransactionManager<>(producerFactory);
    }

    @Bean
    NewTopic createWithdrawTopic() {
        return TopicBuilder.name(withdrawTopicName).partitions(3).replicas(3).build();
    }

    @Bean
    NewTopic createDepositTopic() {
        return TopicBuilder.name(depositTopicName).partitions(3).replicas(3).build();
    }
}
```

And now I can refer to this bin by this specific name.

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

TransfersController.java | *TransferServiceImpl.java | KafkaConfig.java

```java
        this.restTemplate = restTemplate;
    }

    @Transactional("kafkaTransactionManager")
    @Override
    public boolean transfer(TransferRestModel transferRestModel) {
        WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());
        DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
                transferRestModel.getRecepientId(), transferRestModel.getAmount());

        try {
            kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
                    withdrawalEvent);
            LOGGER.info("Sent event to withdrawal topic.");

            // Business logic that causes and error
            callRemoteServce();

            kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
            LOGGER.info("Sent event to deposit topic");

        } catch (Exception ex) {
            LOGGER.error(ex.getMessage(), ex);
            throw new TransferServiceException(ex);
        }

        return true;
    }

    private ResponseEntity<String> callRemoteServce() throws Exception {
        String requestUrl = "http://localhost:8082/response/200";
        ResponseEntity<String> response = restTemplate.exchange(requestUrl, HttpMethod.GET, null, String.class);

        if (response.getStatusCode().value() == HttpStatus.SERVICE_UNAVAILABLE.value()) {
            throw new Exception("Destination Microservice not availble");
        }
```

Let's save changes to Kafka config class and go back to Transfer service implementation.

126. Creating Jpa Transaction Manager

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/KafkaConfig.java - Spring Tool Suite 4

TransfersController.java | TransferServiceImpl.java | *KafkaConfig.java

```java
        props.put(ProducerConfig.ENABLE_IDEMPOTENCE_CONFIG, idempotence);
        props.put(ProducerConfig.MAX_IN_FLIGHT_REQUESTS_PER_CONNECTION, inflightRequests);

        props.put(ProducerConfig.TRANSACTIONAL_ID_CONFIG, transactionalIdPrefix);

        return props;
    }

    @Bean
    ProducerFactory<String, Object> producerFactory() {
        return new DefaultKafkaProducerFactory<>(producerConfigs());
    }

    @Bean
    KafkaTemplate<String, Object> kafkaTemplate(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTemplate<>(producerFactory);
    }

    @Bean("kafkaTransactionManager")
    KafkaTransactionManager<String, Object> kafkaTransactionManager(ProducerFactory<String, Object> producerFactory) {
        return new KafkaTransactionManager<>(producerFactory);
    }

    @Bean("transactionManager")
    JpaTransactionManager jpaTransactionManager(EntityManagerFactory entityManagerFactory) {
        return new JpaTransactionManager(entityManagerFactory);
    }

    @Bean
    NewTopic createWithdrawTopic() {
        return TopicBuilder.name(withdrawTopicName).partitions(3).replicas(3).build();
    }

    @Bean
    NewTopic createDepositTopic() {
        return TopicBuilder.name(
    }
}
```

So now that I have transaction manager that Spring Framework can use for GPA transactions I can try

1.25x 5:09 / 5:19

127. Synchronized Transaction: Saving to database

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

TransfersController.java | TransferServiceImpl.java | KafkaConfig.java

```
38      }
39
40      @Transactional("kafkaTransactionManager")
41      @Override
42      public boolean transfer(TransferRestModel transferRestModel) {
43          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
44                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
45          DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
46                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
47
48          TransferEntity transferEntity = new TransferEntity();
49          BeanUtils.copyProperties(transferRestModel, transferEntity);
50          transferEntity.setTransferId(Uuid.randomUuid().toString());
51
52          try {
53
54              // Save record to a database table
55              transferRepository.save(transferEntity);
56
57              kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
58                      withdrawalEvent);
59              LOGGER.info("Sent event to withdrawal topic.");
60
61              // Business logic that causes and error
62              callRemoteServce();
63
64              kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
65              LOGGER.info("Sent event to deposit topic");
66
67          } catch (Exception ex) {
68              LOGGER.error(ex.getMessage(), ex);
69              throw new TransferServiceException(ex);
70          }
71
72          return true;
73      }
74
```

And now that we updated this method to work with the database
we will also need to update transactional

1.25x 3:37 / 4:04

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

TransfersController.java | *TransferServiceImpl.java | KafkaConfig.java

```
38      }
39
40      @Transactional("transactionManager")
41      @Override
42      public boolean transfer(TransferRestModel transferRestModel) {
43          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestModel.getSenderId(),
44                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
45          DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.getSenderId(),
46                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
47
48          TransferEntity transferEntity = new TransferEntity();
49          BeanUtils.copyProperties(transferRestModel, transferEntity);
50          transferEntity.setTransferId(Uuid.randomUuid().toString());
51
52          try {
53
54              // Save record to a database table
55              transferRepository.save(transferEntity);
56
57              kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-money-topic"),
58                      withdrawalEvent);
59              LOGGER.info("Sent event to withdrawal topic.");
60
61              // Business logic that causes and error
62              callRemoteServce();
63
64              kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-topic"), depositEvent);
65              LOGGER.info("Sent event to deposit topic");
66
67          } catch (Exception ex) {
68              LOGGER.error(ex.getMessage(), ex);
69              throw new TransferServiceException(ex);
70          }
71
72          return true;
73      }
74
```

So let's save changes and let's continue to the next lesson.

128. Enable logging for Kafka and Jpa Transaction Managers

```
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringS
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serialize
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction=TRACE
logging.level.org.springframework.transaction=TRACE
logging.level.org.springframework.orgm.jpa.JpaTransactionManager=DEBUG

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

Actually, if all login levels of all these packages are set to trace, then there is a lot more login

1:03 / 3:00

128. Enable logging for Kafka and Jpa Transaction Managers

```
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringS
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serialize
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction=DEBUG
logging.level.org.springframework.transaction=DEBUG
logging.level.org.springframework.orgm.jpa.JpaTransactionManager=DEBUG

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
```

Now when it comes to GPA transaction manager, I enabled login level for the class itself and when it

1:17 / 3:00

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Mon 26 Aug 4:01 PM

128. Enable logging for Kafka and Jpa Transaction Managers

payments - TransferService/src/main/resources/application.properties - Spring Tool Suite 4

Package Explorer

- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
  - src/main/java
    - com.appsdeveloperblog.estore.transfers
    - com.appsdeveloperblog.estore.transfers.error
    - com.appsdeveloperblog.estore.transfers.io
    - com.appsdeveloperblog.estore.transfers.model
    - com.appsdeveloperblog.estore.transfers.servic
    - com.appsdeveloperblog.estore.transfers.ui
  - src/main/resources
    - static
    - templates
    - application.properties
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd

Boot Dashboard

- local
  - core
  - DepositService
  - mockservice
  - TransferService
  - WithdrawalService

4 elements hidden by filter

application.properties

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction.KafkaTransactionManager=DEBUG
logging.level.org.springframework.transaction=DEBUG
logging.level.org.springframework.orm.jpa.JpaTransactionManager=DEBUG

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

All right, so when your Java method does not work with GPA but works with Kafka, you can use transactional

Pause

1.25x 1:50 / 3:00

Chrome File Edit View History Bookmarks Profiles Tab Window Help 42% Mon 26 Aug 4:02 PM

128. Enable logging for Kafka and Jpa Transaction Managers

payments - TransferService/src/main/resources/application.properties - Spring Tool Suite 4

Package Explorer

- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
  - src/main/java
    - com.appsdeveloperblog.estore.transfers
    - com.appsdeveloperblog.estore.transfers.error
    - com.appsdeveloperblog.estore.transfers.io
    - com.appsdeveloperblog.estore.transfers.model
    - com.appsdeveloperblog.estore.transfers.servic
    - com.appsdeveloperblog.estore.transfers.ui
  - src/main/resources
    - static
    - templates
    - application.properties
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd

Boot Dashboard

- local
  - core
  - DepositService
  - mockservice
  - TransferService
  - WithdrawalService

4 elements hidden by filter

application.properties

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction.KafkaTransactionManager=DEBUG
logging.level.org.springframework.transaction=DEBUG
logging.level.org.springframework.orm.jpa.JpaTransactionManager=DEBUG
logging.level.org.apache.kafka.clients.producer.internals.TransactionManager=DEBUG

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

All right, so now I can save changes to my application properties file, and I can run my applications

Pause

1.25x 2:55 / 3:00

129. Trying how synchronized transactions work

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000

spring.kafka.producer.properties.max.in.flight.requests.per.connection=5
spring.kafka.producer.properties.enable.idempotence=true

withdraw-money-topic=withdraw-money-topic
deposit-money-topic=deposit-money-topic

spring.kafka.producer.transaction-id-prefix=transfer-service-${random.value}-

logging.level.org.springframework.kafka.transaction.KafkaTransactionManager=DEBUG
logging.level.org.springframework.transaction=DEBUG
logging.level.org.springframework.orm.jpa.JpaTransactionManager=DEBUG
logging.level.org.apache.kafka.clients.producer.internals.TransactionManager=DEBUG

spring.datasource.username=sergey
spring.datasource.password=sergey
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

So I'll go back to my development environment, expand projects on Spring Boot dashboard and will look

129. Trying how synchronized transactions work

http://localhost:54685/transfers

POST http://localhost:54685/transfers

```
{
    "senderId":"123",
    "recepientId":"124",
    "amount":0
}
```

Status: 200 OK Time: 655 ms Size: 168 B

```
true
```

The request is successful.

129. Trying how synchronized transactions work

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000
```

```
2024-04-01T09:55:00.705-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.711-04:00  INFO 20954 --- [4a7c656480ef4-0] org.apache.kafka.clients.Metadata
2024-04-01T09:55:00.712-04:00  INFO 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.826-04:00  INFO 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.826-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.827-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.864-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.865-04:00  INFO 20954 --- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImp
2024-04-01T09:55:00.866-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.871-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.878-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:00.909-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.005-04:00  INFO 20954 --- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImp
2024-04-01T09:55:01.012-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.012-04:00  INFO 20954 --- [o-auto-1-exec-2] c.a.e.t.service.TransferServiceImp
2024-04-01T09:55:01.012-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.012-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.s.orm.jpa.JpaTransactionManager
2024-04-01T09:55:01.013-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.s.orm.jpa.JpaTransactionManager
2024-04-01T09:55:01.020-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.021-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.031-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.038-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.039-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.044-04:00 DEBUG 20954 --- [4a7c656480ef4-0] o.a.k.c.p.internals.TransactionMan
2024-04-01T09:55:01.045-04:00 DEBUG 20954 --- [o-auto-1-exec-2] o.s.orm.jpa.JpaTransactionManager
```

1.25x 3:08 / 10:59

129. Trying how synchronized transactions work

Find: COMMITTIN

Replace with:

Direction: Forward / Backward

Options: Case sensitive, Wrap search, Whole word, Incremental, Regular expressions

Find | Select All | Replace/Find | Replace | Replace All | Close

COMMITTING_TRANSACTION

```
2024-04-01T09:52:05.434-04:00  INFO 20954 --- [          main] com.zaxxer.hikari.HikariDataSource
2024-04-01T09:52:05.435-04:00  INFO 20954 --- [          main] com.zaxxer.hikari.HikariDataSource
2024-04-01T09:52:05.444-04:00  INFO 20954 --- [          main] o.s.b.a.h2.H2ConsoleAutoConfigurat
2024-04-01T09:52:05.537-04:00  INFO 20954 --- [          main] o.hibernate.jpa.internal.util.LogH
2024-04-01T09:52:05.582-04:00  INFO 20954 --- [          main] org.hibernate.Version
2024-04-01T09:52:05.608-04:00  INFO 20954 --- [          main] o.h.c.internal.RegionFactoryInitia
2024-04-01T09:52:05.792-04:00  INFO 20954 --- [          main] o.s.o.j.p.SpringPersistenceUnitInf
```

1.25x 3:38 / 10:59

129. Trying how synchronized transactions work

Find: COMMITTING_TRANSACTION

Replace with:

Direction: Forward / Backward

Scope: All / Selected lines

Options: Case sensitive, Wrap search, Whole word, Incremental, Regular expressions

Find, Select All, Replace/Find, Replace, Replace All, Close

```
556480ef4-0] Enqueuing transactional request AddPartitionsToTxnRequestData(transactions=[], v3AndB
556480ef4-0] Successfully added partitions [withdraw-money-topic-2] to transaction
556480ef4-0] ProducerId of partition withdraw-money-topic-2 set to 32004 with epoch 0. Reinitializ
556480ef4-0] ProducerId: 32004; Set last ack'd sequence number for topic-partition withdraw-money-

556480ef4-0] Begin adding new partition deposit-money-topic-1 to transaction

556480ef4-0] Enqueuing transactional request AddPartitionsToTxnRequestData(transactions=[], v3AndB


556480ef4-0] Successfully added partitions [deposit-money-topic-1] to transaction
556480ef4-0] ProducerId of partition deposit-money-topic-1 set to 32004 with epoch 0. Reinitialize
556480ef4-0] ProducerId: 32004; Set last ack'd sequence number for topic-partition deposit-money-t
556480ef4-0] Transition from state IN_TRANSACTION to COMMITTING_TRANSACTION
556480ef4-0] Enqueuing transactional request EndTxnRequestData(transactionalId='transfer-service-f
556480ef4-0] Transition from state COMMITTING_TRANSACTION to READY
```

Transitioning from state in transaction to committing.

1.25x 3:55 / 10:59

129. Trying how synchronized transactions work

Find: COMMITTING_TRANSACTION

Replace with:

Direction: Forward / Backward

Scope: All / Selected lines

Options: Case sensitive, Wrap search, Whole word, Incremental, Regular expressions

Find, Select All, Replace/Find, Replace, Replace All, Close

```
xec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to withdrawal topic.
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] c.a.e.t.service.TransferServiceImpl      : Received response from mock service: 200
xec-2] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] c.a.e.t.service.TransferServiceImpl      : Sent event to deposit topic
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] o.s.orm.jpa.JpaTransactionManager        : Initiating transaction commit
xec-2] o.s.orm.jpa.JpaTransactionManager        : Committing JPA transaction on EntityManager [Ses
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
ef4-0] o.a.k.c.p.internals.TransactionManager   : [Producer clientId=producer-transfer-service-f70
xec-2] o.s.orm.jpa.EntityManagerFactoryUtils    : Closing pre-bound JPA EntityManager after tr
```

Now let's look for messages from transaction manager.

1.25x 4:22 / 10:59

129. Trying how synchronized transactions work

Found thread bound entity and then it says participating in existing transaction.

1.25x 5:02 / 10:59

129. Trying how synchronized transactions work

And finally it says initiating transaction commit and then says committing a transaction on entity manager.

1.25x 5:11 / 10:59

129. Trying how synchronized transactions work

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000
```

```
dlingDeserializer

pInfoParser      : Kafka version: 3.6.1
pInfoParser      : Kafka commitId: 5e3c2b738d253ff5
pInfoParser      : Kafka startTimeMs: 1711979577643
edKafkaConsumer  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
eApplication     : Started WithdrawalServiceApplication in 1.409 seconds (process running for 1.84
.Metadata        : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
mmerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
```

Now this consumer microservice is configured to read only those messages that were successfully committed.

1.25x 5:46 / 10:59

129. Trying how synchronized transactions work

```
server.port=0
spring.kafka.producer.bootstrap-servers=localhost:9092
spring.kafka.producer.key-serializer=org.apache.kafka.common.serialization.StringSerializer
spring.kafka.producer.value-serializer=org.springframework.kafka.support.serializer.JsonSeri
spring.kafka.producer.acks=all

spring.kafka.producer.properties.delivery.timeout.ms=120000
spring.kafka.producer.properties.linger.ms=0
spring.kafka.producer.properties.request.timeout.ms=30000
```

```
pache.kafka.clients.Metadata          : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.t.service.TransferServiceImpl        : Sent event to withdrawal topic.
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.t.service.TransferServiceImpl        : Received response from mock service: 200
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.t.service.TransferServiceImpl        : Sent event to deposit topic
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
rm.jpa.JpaTransactionManager          : Initiating transaction commit
rm.jpa.JpaTransactionManager          : Committing JPA transaction on EntityManager [SessionImpl(23
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
.c.p.internals.TransactionManager     : [Producer clientId=producer-transfer-service-f701e7af25800f
rm.jpa.JpaTransactionManager          : Not closing pre-bound JPA EntityManager after transaction
pa                                    =producer-transfer-service-f701e7af25800f
pa                                    =producer-transfer-service-f701e7af25800f
```

I will first look up at the port number of transfer microservice.

1.25x 6:10 / 10:59

129. Trying how synchronized transactions work

http://localhost:54685/h2-console

Private Browsing

Safari is designed with privacy in mind by preventing tracking by default. Private Browsing adds additional privacy protections for all tabs in this window. After you close this window, Safari won't remember the pages you visited, your search history, or your AutoFill information.

In the browser window I will go to localhost, then port number 54685 forward slash and then the path

1.25x 6:28 / 10:59

129. Trying how synchronized transactions work

SELECT * FROM TRANSFERS

AMOUNT	RECEPIENT_ID	SENDER_ID	TRANSFER_ID
0.00	124	123	R-4AeW_BR5WdaltvL6yJ_A

(1 row, 3 ms)

So database transaction was also successful.

1.5x 7:26 / 10:59

Chrome File Edit View History Bookmarks Profiles Tab Window Help 39% Mon 26 Aug 4:09 PM

SpringToolSuite4 File Edit Source Refactor Navigate Search Project Run Window Help Mon Apr 1 10:03 AM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

Package Explorer
- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
- WithdrawalService [boot]

application.properties | TransferServiceImpl.java

```
40●     @Transactional("transactionManager")
41      @Override
42      public boolean transfer(TransferRestModel transferRestModel) {
43          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestM
44                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
45          DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.get
46                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
47
48          TransferEntity transferEntity = new TransferEntity();
49          BeanUtils.copyProperties(transferRestModel, transferEntity);
50          transferEntity.setTransferId(Uuid.randomUuid().toString());
51
52          try {
53
54              // Save record to a database table
55              transferRepository.save(transferEntity);
56
57              kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-mon
58                      withdrawalEvent);
59              LOGGER.info("Sent event to withdrawal topic.");
60
61              // Business logic that causes and error
62              callRemoteServce();
63
64              kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money
65              LOGGER.info("Sent event to deposit topic");
66
67          } catch (Exception ex) {
```

Boot Dashboard
- local
  - core
  - DepositService [:54726]
  - mockservice [:8082]
  - TransferService [:54685]
  - WithdrawalService [:54743]

4 elements hidden by filter

Problems | Javadoc | Declaration | Console

```
dlingDeserializer

pInfoParser      : Kafka version: 3.6.1
pInfoParser      : Kafka commitId: 5e3c2b738d253ff5
pInfoParser      : Kafka startTimeMs: 1711979577643
dedKafkaConsumer : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
eApplication     : Started WithdrawalServiceApplication in 1.409 seconds (process running for 1.84
```

In this microservice I will simply stop microservice that it is trying to send a HTTP request to mock

Udemy

Chrome File Edit View History Bookmarks Profiles Tab Window Help 39% Mon 26 Aug 4:09 PM

SpringToolSuite4 File Edit Source Refactor Navigate Search Project Run Window Help Mon Apr 1 10:03 AM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

Package Explorer
- core [boot]
- DepositService [boot]
- mockservice [boot]
- TransferService [boot]
- WithdrawalService [boot]

application.properties | TransferServiceImpl.java

```
40●     @Transactional("transactionManager")
41      @Override
42      public boolean transfer(TransferRestModel transferRestModel) {
43          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestM
44                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
45          DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.get
46                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
47
48          TransferEntity transferEntity = new TransferEntity();
49          BeanUtils.copyProperties(transferRestModel, transferEntity);
50          transferEntity.setTransferId(Uuid.randomUuid().toString());
51
52          try {
53
54              // Save record to a database table
55              transferRepository.save(transferEntity);
56
57              kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-mon
58                      withdrawalEvent);
59              LOGGER.info("Sent event to withdrawal topic.");
60
61              // Business logic that causes and error
62              callRemoteServce();
63
64              kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money
65              LOGGER.info("Sent event to deposit topic");
66
67          } catch (Exception ex) {
```

Boot Dashboard
- local
  - core
  - DepositService [:54726]
  - mockservice [:8082]
  - TransferService [:54685]
  - WithdrawalService [:54743]

4 elements hidden by filter

Problems | Javadoc | Declaration | Console

```
dlingDeserializer

pInfoParser      : Kafka version: 3.6.1
pInfoParser      : Kafka commitId: 5e3c2b738d253ff5
pInfoParser      : Kafka startTimeMs: 1711979577643
dedKafkaConsumer : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
eApplication     : Started WithdrawalServiceApplication in 1.409 seconds (process running for 1.84
```

And to do that I will select this mock service from the Springboot dashboard and we'll click on this

Udemy

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

```java
@Transactional("transactionManager")
@Override
public boolean transfer(TransferRestModel transferRestModel) {
    WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestM
            transferRestModel.getRecepientId(), transferRestModel.getAmount());
    DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.get
            transferRestModel.getRecepientId(), transferRestModel.getAmount());

    TransferEntity transferEntity = new TransferEntity();
    BeanUtils.copyProperties(transferRestModel, transferEntity);
    transferEntity.setTransferId(Uuid.randomUuid().toString());

    try {

        // Save record to a database table
        transferRepository.save(transferEntity);

        kafkaTemplate.send(environment.getProperty("withdraw-money-topic", "withdraw-mon
                withdrawalEvent);
        LOGGER.info("Sent event to withdrawal topic.");

        // Business logic that causes and error
        callRemoteServce();

        kafkaTemplate.send(environment.getProperty("deposit-money-topic", "deposit-money-
        LOGGER.info("Sent event to deposit topic");

    } catch (Exception ex) {
```

```
dlingDeserializer
pInfoParser     : Kafka version: 3.6.1
pInfoParser     : Kafka commitId: 5e3c2b738d253ff5
pInfoParser     : Kafka startTimeMs: 1711979577643
dedKafkaConsumer : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
eApplication    : Started WithdrawalServiceApplication in 1.409 seconds (process running for 1.84
```

And to do that I will select this mock service from the Springboot dashboard and we'll click on this

Postman — POST http://localhost:54685/transfers

```
POST http://localhost:54685/transfers
```

```
{
    "senderId":"123",
    "recepientId":"124",
    "amount":10
}
```

Status: 500 Internal Server Error Time: 159 ms Size: 264 B

```
{
    "timestamp": "2024-04-01T14:04:46.705+00:00",
    "status": 500,
    "error": "Internal Server Error",
```

So let's check log file and see if there is a message about transaction rollback.

I will search for messages from GBA Transaction Manager.

It found a message that says initiating transaction rollback.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 39% Mon 26 Aug 4:10 PM

Safari File Edit View History Bookmarks Develop Window Help Mon Apr 1 10:06 AM

Private localhost

Auto commit Max rows: 1000 Auto complete Off Auto select On

jdbc:h2:mem:testdb
TRANSFERS
AMOUNT
RECEPIENT_ID
SENDER_ID
TRANSFER_ID
Indexes
INFORMATION_SCHEMA
Users
H2 2.2.224 (2023-09-17)

Run Run Selected Auto complete Clear SQL statement:

```
SELECT * FROM TRANSFERS
```

SELECT * FROM TRANSFERS;

AMOUNT	RECEPIENT_ID	SENDER_ID	TRANSFER_ID
0.00	124	123	R-4AeW_BR5WdaftvL6yJ_A

(1 row, 1 ms)

Edit

Udemy

And I don't have a new record.

Chrome File Edit View History Bookmarks Profiles Tab Window Help 39% Mon 26 Aug 4:10 PM

SpringToolSuite4 File Edit Source Refactor Navigate Search Project Run Window Help Mon Apr 1 10:06 AM

payments - TransferService/src/main/java/com/appsdeveloperblog/estore/transfers/service/TransferServiceImpl.java - Spring Tool Suite 4

Package Explorer
core [boot]
DepositService [boot]
mockservice [boot]
TransferService [boot]
WithdrawalService [boot]

application.properties TransferServiceImpl.java

```
40●     @Transactional("transactionManager")
41      @Override
42      public boolean transfer(TransferRestModel transferRestModel) {
43          WithdrawalRequestedEvent withdrawalEvent = new WithdrawalRequestedEvent(transferRestI
44                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
45          DepositRequestedEvent depositEvent = new DepositRequestedEvent(transferRestModel.get!
46                  transferRestModel.getRecepientId(), transferRestModel.getAmount());
47
48          TransferEntity transferEntity = new TransferEntity();
49          BeanUtils.copyProperties(transferRestModel, transferEntity);
50          transferEntity.setTransferId(Uuid.randomUuid().toString());
```

Problems Javadoc Declaration Console

```
pInfoParser      : Kafka version: 3.6.1
pInfoParser      : Kafka commitId: 5e3c2b738d253ff5
pInfoParser      : Kafka startTimeMs: 1711979577643
dedKafkaConsumer : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
eApplication     : Started WithdrawalServiceApplication in 1.409 seconds (process running for 1.84
.Metadata        : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
umerCoordinator  : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
enerContainer    : amount-widthrawal-event: partitions assigned: [withdraw-money-topic-0, withdraw
stedEventHandler : Received a new withdrawal event: 0
.NetworkClient   : [Consumer clientId=consumer-amount-widthrawal-event-1, groupId=amount-widthrawa
```

Boot Dashboard
local
core
DepositService [:54726]
mockservice
TransferService [:54685]
WithdrawalService [:54743]

4 elements hidden by filter

Udemy

If I scroll down and I see that it did not receive a new event with an amount of $10.