|  |  |
| --- | --- |
| **Topic** | Apache Kafka |
| **Document Name** | KAFKA-EX-03 |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Difficulty Level** | | | |
| **Beginner** | **Junior** | **Senior** | **Expert** |
| □ | ■ | □ | □ |

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Ver | Comments |
| 05.03.2025 | Mennan Tekbir | 1.0 | Initial Draft |
| 05.03.2025 | Mehmet Erdem Önal | 1.1 | Revisions |

# Apache Kafka

## Exercise KAFKA-EX-03:

**Definiton:** Create 2 sample java programs; one for Producer one for Consumer. Publish java objects ( you can use following class ) from Producer app. Read those objects from Consumer app and print them to the console.

**Java Classes & Objects:** <https://www.geeksforgeeks.org/classes-objects-java>

**Java Program For Class Example:**

class Student {

  // data member (also instance variable)

  int id;

  // data member (also instance variable)

  String name;

  public static void main(String args[])

  {

    // creating an object of

    // Student

    Student s1 = new Student();

    System.out.println(s1.id);

    System.out.println(s1.name);

 }

}

**Sample Kafka Producer code:** <https://github.com/chrissearle/kafka-java-to-scala/blob/master/java-starter/producer/src/main/java/net/chrissearle/kafka/BasicProducer.java>

**Sample Kafka Consumer code:** <https://github.com/chrissearle/kafka-java-to-scala/blob/master/java-starter/consumer/src/main/java/net/chrissearle/kafka/BasicConsumer.java>

Please provide screenshots to show your work.

## KAFKA-EX-03 Solution:

**Your Answer:metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.metin, ekran görüntüsü, yazı tipi, siyah beyaz içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.**

**ahmetilkerozturk@kafka-vm:~/kafka-java-ex03$ cd ~/kafka-java-ex03**

**java -cp ".:libs/\*" ProducerExample**

**[main] INFO org.apache.kafka.clients.producer.ProducerConfig - ProducerConfig values:**

**acks = -1**

**auto.include.jmx.reporter = true**

**batch.size = 16384**

**bootstrap.servers = [35.205.10.107:9092]**

**buffer.memory = 33554432**

**client.dns.lookup = use\_all\_dns\_ips**

**client.id = producer-1**

**compression.type = none**

**connections.max.idle.ms = 540000**

**delivery.timeout.ms = 120000**

**enable.idempotence = true**

**enable.metrics.push = true**

**interceptor.classes = []**

**key.serializer = class org.apache.kafka.common.serialization.StringSerializer**

**linger.ms = 0**

**max.block.ms = 60000**

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

**max.request.size = 1048576**

**metadata.max.age.ms = 300000**

**metadata.max.idle.ms = 300000**

**metric.reporters = []**

**metrics.num.samples = 2**

**metrics.recording.level = INFO**

**metrics.sample.window.ms = 30000**

**partitioner.adaptive.partitioning.enable = true**

**partitioner.availability.timeout.ms = 0**

**partitioner.class = null**

**partitioner.ignore.keys = false**

**receive.buffer.bytes = 32768**

**reconnect.backoff.max.ms = 1000**

**reconnect.backoff.ms = 50**

**request.timeout.ms = 30000**

**retries = 2147483647**

**retry.backoff.max.ms = 1000**

**retry.backoff.ms = 100**

**sasl.client.callback.handler.class = null**

**sasl.jaas.config = null**

**sasl.kerberos.kinit.cmd = /usr/bin/kinit**

**sasl.kerberos.min.time.before.relogin = 60000**

**sasl.kerberos.service.name = null**

**sasl.kerberos.ticket.renew.jitter = 0.05**

**sasl.kerberos.ticket.renew.window.factor = 0.8**

**sasl.login.callback.handler.class = null**

**sasl.login.class = null**

**sasl.login.connect.timeout.ms = null**

**sasl.login.read.timeout.ms = null**

**sasl.login.refresh.buffer.seconds = 300**

**sasl.login.refresh.min.period.seconds = 60**

**sasl.login.refresh.window.factor = 0.8**

**sasl.login.refresh.window.jitter = 0.05**

**sasl.login.retry.backoff.max.ms = 10000**

**sasl.login.retry.backoff.ms = 100**

**sasl.mechanism = GSSAPI**

**sasl.oauthbearer.clock.skew.seconds = 30**

**sasl.oauthbearer.expected.audience = null**

**sasl.oauthbearer.expected.issuer = null**

**sasl.oauthbearer.jwks.endpoint.refresh.ms = 3600000**

**sasl.oauthbearer.jwks.endpoint.retry.backoff.max.ms = 10000**

**sasl.oauthbearer.jwks.endpoint.retry.backoff.ms = 100**

**sasl.oauthbearer.jwks.endpoint.url = null**

**sasl.oauthbearer.scope.claim.name = scope**

**sasl.oauthbearer.sub.claim.name = sub**

**sasl.oauthbearer.token.endpoint.url = null**

**security.protocol = PLAINTEXT**

**security.providers = null**

**send.buffer.bytes = 131072**

**socket.connection.setup.timeout.max.ms = 30000**

**socket.connection.setup.timeout.ms = 10000**

**ssl.cipher.suites = null**

**ssl.enabled.protocols = [TLSv1.2, TLSv1.3]**

**ssl.endpoint.identification.algorithm = https**

**ssl.engine.factory.class = null**

**ssl.key.password = null**

**ssl.keymanager.algorithm = SunX509**

**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.apache.kafka.common.serialization.StringSerializer**

**[main] INFO org.apache.kafka.common.telemetry.internals.KafkaMetricsCollector - initializing Kafka metrics collector**

**[main] INFO org.apache.kafka.clients.producer.KafkaProducer - [Producer clientId=producer-1] Instantiated an idempotent producer.**

**[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka version: 3.7.0**

**[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka commitId: 2ae524ed625438c5**

**[main] INFO org.apache.kafka.common.utils.AppInfoParser - Kafka startTimeMs: 1751463530550**

**[kafka-producer-network-thread | producer-1] INFO org.apache.kafka.clients.Metadata - [Producer clientId=producer-1] Cluster ID: 3TkcRs\_yQqK\_TXuOHG3QbQ**

**[kafka-producer-network-thread | producer-1] INFO org.apache.kafka.clients.producer.internals.TransactionManager - [Producer clientId=producer-1] ProducerId set to 1 with epoch 0**

**[main] INFO org.apache.kafka.clients.producer.KafkaProducer - [Producer clientId=producer-1] Closing the Kafka producer with timeoutMillis = 9223372036854775807 ms.**

**[main] INFO org.apache.kafka.common.metrics.Metrics - Metrics scheduler closed**

**[main] INFO org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.metrics.JmxReporter**

**[main] INFO org.apache.kafka.common.metrics.Metrics - Closing reporter org.apache.kafka.common.telemetry.internals.ClientTelemetryReporter**

**[main] INFO org.apache.kafka.common.metrics.Metrics - Metrics reporters closed**

**[main] INFO org.apache.kafka.common.utils.AppInfoParser - App info kafka.producer for producer-1 unregistered**

**Message sent successfully!**