

**Большие данные**

**613x-010402D x={1,2,3} осень 2025**

**Лабораторная работа №2**

**Обработка данных с использованием Kafka**

Сергей Борисович Попов  
[sepo@ssau.ru](mailto:sepo@ssau.ru)

**Материалы лекций:**

[https://1drv.ms/f/c/5ed33c8b23e26391/EpMzlOhQgt5OqcqHF\\_ChjwB0NP5AovVgqYTfBokEY1zhA](https://1drv.ms/f/c/5ed33c8b23e26391/EpMzlOhQgt5OqcqHF_ChjwB0NP5AovVgqYTfBokEY1zhA)

**Цель:** реализация взаимодействия через платформу Kafka  
источников событий и их обработчиков,  
реализованных на Python

**Варианты реализации:**

1. Один источник (producer), один обработчик (consumer)
2. Один источник (producer), распределяющий события по двум разделам (two partitions), два обработчика (consumer), объединённые в группу, обрабатывающие каждый свой раздел (распределённая обработка)

**Исходные данные:**

файл в формате csv, содержащий описание событий

**Задача обработки:**

для каждого перевозчика (vendor\_id) просуммировать количество поездок, число перевезённых пассажиров (passenger\_count), расстояние поездок (trip\_distance) и сумму оплаты (total\_amount)

# Установка Kafka и простейший сценарий работы

Загружаем бинарный дистрибутив с сайта проекта:

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

```
$ tar -xzf kafka_2.13-3.9.0.tgz  
$ cd kafka_2.13-3.9.0
```

## Kafka with KRaft

```
$ KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"  
  
$ bin/kafka-storage.sh format --standalone -t $KAFKA_CLUSTER_ID  
-c config/kraft/reconfig-server.properties  
  
$ bin/kafka-server-start.sh config/kraft/reconfig-server.properties
```

## Using JVM Based Apache Kafka Docker Image

## Kafka with ZooKeeper

● ● ● kafka\_2.13-3.9.0 - java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercent=35 -XX:+ExplicitGCInvokesConcurrent -XX:MaxInlineLevel=15 -Djava.awt.headless=true -Xloggc:/Users/sep0/kafka\_2.13-3.9.0/bin./logs/kafka.log

```
(base) sep0@MacBookPro16 kafka_2.13-3.9.0 % KAFKA_CLUSTER_ID="$({bin}/kafka-storage.sh random-uuid)"  
(base) sep0@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-storage.sh format --standalone -t $KAFKA_CLUSTER_ID -c config/kraft/reconfig-server.properties  
Formatting dynamic metadata voter directory /tmp/kraft-combined-logs with metadata.version 3.9-IV0.  
(base) sep0@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-server-start.sh config/kraft/reconfig-server.properties  
[2024-12-17 11:42:00,288] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)  
[2024-12-17 11:42:00,498] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)  
[2024-12-17 11:42:00,699] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.common.utils.LoggingSignalHandler)  
[2024-12-17 11:42:00,702] INFO [ControllerServer id=1] Starting controller (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,204] INFO Updated connection-accept-rate max connection creation rate to 2147483647 (kafka.network.ConnectionQuotas)  
[2024-12-17 11:42:01,250] INFO [SocketServer listenerType=CONTROLLER, nodeId=1] Created data-plane acceptor and processors for endpoint : ListenerName(CONTROLLER) (kafka.network.SocketServer)  
[2024-12-17 11:42:01,259] INFO authorizationStart completed for endpoint CONTROLLER. Endpoint is now READY. (org.apache.kafka.server.network.EndpointReadyFutures)  
[2024-12-17 11:42:01,261] INFO [SharedServer id=1] Starting SharedServer (kafka.server.SharedServer)  
[2024-12-17 11:42:01,337] INFO [LogLoader partition=_cluster_metadata-0, dir=/tmp/kraft-combined-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog$)  
[2024-12-17 11:42:01,338] INFO [LogLoader partition=_cluster_metadata-0, dir=/tmp/kraft-combined-logs] Reloading from producer snapshot and rebuilding producer state from offset 0 (kafka.log.UnifiedLog$)  
[2024-12-17 11:42:01,338] INFO [LogLoader partition=_cluster_metadata-0, dir=/tmp/kraft-combined-logs] Producer state recovery took 0ms for snapshot load and 0ms for segment recovery from offset 0 (kafka.log.UnifiedLog$)  
[2024-12-17 11:42:01,386] INFO Initialized snapshots with IDs SortedSet(OffsetAndEpoch(offset=0, epoch=0)) from /tmp/kraft-combined-logs/_cluster_metadata-0 (kafka.raft.KafkaMetadataLogs)  
[2024-12-17 11:42:01,409] INFO [raft-expiration-reaper]: Starting (kafka.raft.TimingWheelExpirationService$ExpiredOperationReaper)  
[2024-12-17 11:42:01,419] INFO [RaftManager id=1] Starting request manager with bootstrap servers: [localhost:9093 (id: -2 rack: null)] (org.apache.kafka.raft.KafkaRaftClient)  
[2024-12-17 11:42:01,430] INFO [RaftManager id=1] Reading KRaft snapshot and log as part of the initializtion (org.apache.kafka.raft.KafkaRaftClient)  
[2024-12-17 11:42:01,433] INFO [RaftManager id=1] Loading snapshot (OffsetAndEpoch(offset=0, epoch=0)) since log start offset (-0) is greater than the internal listener's next offset (-1) (org.apache.kafka.raft.internals.KRaftControlRecordStateMachine)  
[2024-12-17 11:42:01,442] INFO [RaftManager id=1] Latest kraft.version is KRAFT_VERSION_1 at offset -1 (org.apache.kafka.raft.internals.KRaftControlRecordStateMachine)  
[2024-12-17 11:42:01,443] INFO [RaftManager id=1] Latest set of voters is VoterSet(voters={1=VoterNode(voterKey=ReplicaKey{id=1, directoryId=Optional[4YGRG27aQS-oKouus-IJrw]}, listeners=Endpoints(endpoints={ListenerName(CONTROLLER)=localhost:9093}), supp...:0, max_version=1)}) at offset -1 (org.apache.kafka.raft.internals.KRaftControlRecordStateMachine)  
[2024-12-17 11:42:01,444] INFO [RaftManager id=1] Starting voters are VoterSet(voters={1=VoterNode(voterKey=ReplicaKey{id=1, directoryId=Optional[4YGRG27aQS-oKouus-IJrw]}, listeners=Endpoints(endpoints={ListenerName(CONTROLLER)=localhost:9093}), supporte...max_version=1)})) (org.apache.kafka.raft.KafkaRaftClient)  
[2024-12-17 11:42:01,451] INFO [RaftManager id=1] Attempting durable transition to Unattached(votedKey=null, voters=[1], electionTimeoutMs=1725, highWatermark=Optional.empty) from null (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,624] INFO [RaftManager id=1] Completed transition to Unattached(epoch=0, votedKey=null, voters=[1], electionTimeoutMs=1725, highWatermark=Optional.empty) from null (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,627] INFO [RaftManager id=1] Attempting durable transition to CandidateState(localId=1, localDirectoryId=4YGRG27aQS-oKouus-IJrw, epoch=1, retries=1, voteStates={1=org.apache.kafka.raft.CandidateState$VoterState@664e5dee}, highWatermark...ached(epoch=0, votedKey=null, voters=[1], electionTimeoutMs=1725, highWatermark=Optional.empty) (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,628] INFO [RaftManager id=1] Completed transition to CandidateState(localId=1, localDirectoryId=4YGRG27aQS-oKouus-IJrw, epoch=1, retries=1, voteStates={1=org.apache.kafka.raft.CandidateState$VoterState@664e5dee}, highWatermark=Optional...och=0, votedKey=null, voters=[1], electionTimeoutMs=1725, highWatermark=Optional.empty) (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,635] INFO [RaftManager id=1] Attempting durable transition to Leader(localReplicaKey=ReplicaKey{id=1, directoryId=Optional[4YGRG27aQS-oKouus-IJrw]}, epoch=1, epochStartOffset=0, highWatermark=Optional.empty, voterStates={1=ReplicaStat...[4YGRG27aQS-oKouus-IJrw]}, endOffset=Optional.empty, lastFetchTimestamp=-1, lastCaughtUpTimestamp=-1, hasAcknowledgedLeader=true)) from CandidateState(localId=1, localDirectoryId=4YGRG27aQS-oKouus-IJrw, epoch=1, retries=1, voteStates={1=org.apache.kafka.ra...mark=Optional.empty, electionTimeoutMs=1874) (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,636] INFO [RaftManager id=1] Completed transition to Leader(localReplicaKey=ReplicaKey{id=1, directoryId=Optional[4YGRG27aQS-oKouus-IJrw]}, epoch=1, epochStartOffset=0, highWatermark=Optional.empty, voterStates={1=ReplicaState(replica...QS-oKouus-IJrw]}, endOffset=Optional.empty, lastFetchTimestamp=-1, lastCaughtUpTimestamp=-1, hasAcknowledgedLeader=true)) from CandidateState(localId=1, localDirectoryId=4YGRG27aQS-oKouus-IJrw, epoch=1, retries=1, voteStates={1=org.apache.kafka.raft.Candid...onal.empty, electionTimeoutMs=1874) (org.apache.kafka.raft.QuorumState)  
[2024-12-17 11:42:01,651] INFO [kafka-1-raft-outbound-request-thread]: Starting (org.apache.kafka.raft.KafkaNetworkChannel$SendThread)  
[2024-12-17 11:42:01,652] INFO [kafka-1-raft-io-thread]: Starting (org.apache.kafka.raft.KafkaRaftClientDriver)  
[2024-12-17 11:42:01,676] INFO [MetadataLoader id=1] initializeNewPublishers: the loader is still catching up because we still don't know the high water mark yet. (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,676] INFO [RaftManager id=1] Latest kraft.version is KRAFT_VERSION_1 at offset 1 (org.apache.kafka.raft.internals.KRaftControlRecordStateMachine)  
[2024-12-17 11:42:01,677] INFO [RaftManager id=1] Latest set of voters is VoterSet(voters={1=VoterNode(voterKey=ReplicaKey{id=1, directoryId=Optional[4YGRG27aQS-oKouus-IJrw]}, listeners=Endpoints(endpoints={ListenerName(CONTROLLER)=localhost:9093}), supp...:0, max_version=1)}) at offset 2 (org.apache.kafka.raft.internals.KRaftControlRecordStateMachine)  
[2024-12-17 11:42:01,678] INFO [ControllerServer id=1] Waiting for controller quorum voters future (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,678] INFO [ControllerServer id=1] Finished waiting for controller quorum voters future (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,680] INFO [RaftManager id=1] High watermark set to LogOffsetMetadata(offset=3, metadata=Optional[[segmentBaseOffset=0,relativePositionInSegment=174]]) for the first time for epoch 1 based on indexOfHw 0 and voters [ReplicaState(replic...aQS-oKouus-IJrw)], endOffset=Optional[LogOffsetMetadata(offset=3, metadata=Optional[[segmentBaseOffset=0,relativePositionInSegment=174]]), lastFetchTimestamp=-1, lastCaughtUpTimestamp=-1, hasAcknowledgedLeader=true]) (org.apache.kafka.raft.LeaderState)  
[2024-12-17 11:42:01,688] INFO [RaftManager id=1] Registered the listener org.apache.kafka.image.loader.MetadataLoader@079969509 (org.apache.kafka.raft.KafkaRaftClient)  
[2024-12-17 11:42:01,689] INFO [MetadataLoader id=1] handleLoadSnapshot(00000000000000000000000000000000: incrementing HandleLoadSnapshotCount to 1. (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,694] INFO [MetadataLoader id=1] handleLoadSnapshot(00000000000000000000000000000000: generated a metadata delta between offset -1 and this snapshot in 5128 us. (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,695] INFO [MetadataLoader id=1] maybePublishMetadata(SNAPSHOT): The loader is still catching up because we have loaded up to offset -1, but the high water mark is 3 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,696] INFO [MetadataLoader id=1] maybePublishMetadata(LOG_DELTA): The loader finished catching up to the current high water mark of 3 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,697] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing SnapshotGenerator with a snapshot at offset 2 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,727] INFO [RaftManager id=1] Registered the listener org.apache.kafka.controller.QuorumController$QuorumMetaLogListener@1422035566 (org.apache.kafka.raft.KafkaRaftClient)  
[2024-12-17 11:42:01,736] INFO [controller-1-ThrottledChannelReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,737] INFO [controller-1-ThrottledChannelReaper-Producer]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,738] INFO [controller-1-ThrottledChannelReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,740] INFO [controller-1-ThrottledChannelReaper-ControllerMutation]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,766] INFO [ExpirationReaper-1-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)  
[2024-12-17 11:42:01,782] INFO [ControllerServer id=1] Waiting for the controller metadata publishers to be installed (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,782] INFO [ControllerServer id=1] Finished waiting for the controller metadata publishers to be installed (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,782] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing KRaftMetadataCachePublisher with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,782] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing FeaturesPublisher with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,782] INFO [SocketServer listenerType=CONTROLLER, nodeId=1] Enabling request processing. (kafka.network.SocketServer)  
[2024-12-17 11:42:01,782] INFO [ControllerServer id=1] Loaded new metadata Features(metadataVersion=3.9-IV0, finalizedFeatures=21, finalizedFeaturesEpoch=6). (org.apache.kafka.metadata.publisher.FeaturesPublisher)  
[2024-12-17 11:42:01,782] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing ControllerRegistrationsPublisher with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,783] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing ControllerRegistrationManager with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,783] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing DynamicConfigPublisher controller id=1 with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,784] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing DynamicClientQuotaPublisher controller id=1 with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,786] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing ScramPublisher controller id=1 with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,788] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing DelegationTokenPublisher controller id=1 with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,790] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing ControllerMetadataMetricsPublisher with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,791] INFO [MetadataLoader id=1] InitializeNewPublishers: initializing ACPublisher controller id=1 with a snapshot at offset 6 (org.apache.kafka.image.loader.MetadataLoader)  
[2024-12-17 11:42:01,796] INFO Awaiting socket connections on 0.0.0.0:9093. (kafka.network.DataPlaneAcceptor)  
[2024-12-17 11:42:01,809] INFO [controller-1-to-controller-registration-channel-manager]: Starting (kafka.server.NodeToControllerRequestThread)  
[2024-12-17 11:42:01,810] INFO [ControllerServer id=1] Waiting for all of the authorizer futures to be completed (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,810] INFO [ControllerServer id=1] Finished waiting for all of the authorizer futures to be completed (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,810] INFO [ControllerServer id=1] Waiting for all of the SocketServer Acceptors to be started (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,810] INFO [ControllerServer id=1] Finished waiting for all of the SocketServer Acceptors to be started (kafka.server.ControllerServer)  
[2024-12-17 11:42:01,810] INFO [ControllerRegistrationManager id=1] incarnation=26NTWptBQiayjRc1azOK1Q initialized channel manager. (kafka.server.ControllerRegistrationManager)  
[2024-12-17 11:42:01,810] INFO [controller-1-to-controller-registration-channel-manager]: Recorded new KRaft controller, from now on will use node localhost:9093 (id: 1 rack: null) (kafka.server.NodeToControllerRequestThread)  
[2024-12-17 11:42:01,811] INFO [BrokerServer id=1] Transition from SHUTDOWN STARTING (kafka.server.BrokerServer)  
[2024-12-17 11:42:01,812] INFO [ControllerServer id=1] Starting broker (kafka.server.BrokerServer)  
[2024-12-17 11:42:01,812] INFO [ControllerRegistrationManager id=1] incarnation=26NTWptBQiayjRc1azOK1Q sendControllerRegistration: attempting to send ControllerRegistrationRequestData(controllerId=1, incarnationId=26NTWptBQiayjRc1azOK1Q, zkMigrationReady=...=localhost), port=9093, securityProtocol=0), features=[Feature(name='kraft.version', minSupportedVersion=0, maxSupportedVersion=1), Feature(name='metadata.version', minSupportedVersion=1, maxSupportedVersion=21]) (kafka.server.ControllerRegistrationManager)  
[2024-12-17 11:42:01,819] INFO [broker-1-ThrottledChannelReaper-Fetch]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,819] INFO [broker-1-ThrottledChannelReaper-Producer]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,819] INFO [broker-1-ThrottledChannelReaper-Request]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,820] INFO [broker-1-ThrottledChannelReaper-ControllerMutation]: Starting (kafka.server.ClientQuotaManager$ThrottledChannelReaper)  
[2024-12-17 11:42:01,854] INFO [BrokerServer id=1] Waiting for controller quorum voters future (kafka.server.BrokerServer)  
[2024-12-17 11:42:01,854] INFO [BrokerServer id=1] Finished waiting for controller quorum voters future (kafka.server.BrokerServer)  
[2024-12-17 11:42:01,856] INFO [broker-1-to-controller-forwarding-channel-manager]: Starting (kafka.server.NodeToControllerRequestThread)  
[2024-12-17 11:42:01,856] INFO [broker-1-to-controller-forwarding-channel-manager]: Recorded new KRaft controller, from now on will use node localhost:9093 (id: 1 rack: null) (kafka.server.NodeToControllerRequestThread)
```



bin



config



libs



licenses



logs



site-docs



LICENSE



NOTICE



windows



connect-distributed.sh



connect-mirror-maker.sh



connect-plugin-path.sh



connect-standalone.sh



kafka-acls.sh



kafka-broker-api-versions.sh



kafka-client-metrics.sh



kafka-cluster.sh



kafka-configs.sh



kafka-console-consumer.sh



kafka-console-producer.sh



kafka-consumer-groups.sh



kafka-consumer-perf-test.sh



kafka-delegation-tokens.sh



kafka-delete-records.sh



kafka-dump-log.sh

```
(kafka.server.KafkaConfig)
[2024-12-17 11:42:02,179] INFO [BrokerServer id=1] Waiting for the broker to be unfenced (kafka.server.BrokerServ
[2024-12-17 11:42:02,212] INFO [BrokerLifecycleManager id=1] The broker has been unfenced. Transitioning from REC
[2024-12-17 11:42:02,212] INFO [BrokerServer id=1] Finished waiting for the broker to be unfenced (kafka.server.B
[2024-12-17 11:42:02,213] INFO authorizerStart completed for endpoint PLAINTEXT. Endpoint is now READY. (org.apac
[2024-12-17 11:42:02,213] INFO [SocketServer listenerType=BROKER, nodeId=1] Enabling request processing. (kafka.n
[2024-12-17 11:42:02,213] INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.DataPlaneAcceptor)
[2024-12-17 11:42:02,214] INFO [BrokerServer id=1] Waiting for all of the authorizer futures to be completed (kaf
[2024-12-17 11:42:02,214] INFO [BrokerServer id=1] Finished waiting for all of the authorizer futures to be compl
[2024-12-17 11:42:02,214] INFO [BrokerServer id=1] Waiting for all of the SocketServer Acceptors to be started (k
[2024-12-17 11:42:02,214] INFO [BrokerServer id=1] Finished waiting for all of the SocketServer Acceptors to be s
[2024-12-17 11:42:02,214] INFO [BrokerServer id=1] Transition from STARTING to STARTED (kafka.server.BrokerServer
[2024-12-17 11:42:02,214] INFO Kafka version: 3.9.0 (org.apache.kafka.common.utils.AppInfoParser)
[2024-12-17 11:42:02,215] INFO Kafka commitId: a60e31147e6b01ee (org.apache.kafka.common.utils.AppInfoParser)
[2024-12-17 11:42:02,215] INFO Kafka startTimeMs: 1734421322214 (org.apache.kafka.common.utils.AppInfoParser)
[2024-12-17 11:42:02,216] INFO [KafkaRaftServer nodeId=1] Kafka Server started (kafka.server.KafkaRaftServer)
```

Создаём топик:

```
$ bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server
localhost:9092
```

Пишем сообщения в топик:

```
$ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server
localhost:9092
>This is my first event
>This is my second event
```

Читаем сообщения из топика:

```
$ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning
--bootstrap-server localhost:9092
This is my first event
This is my second event
```

● ○ ● kafka\_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:Ma...

```
Last login: Tue Dec 17 11:38:54 on ttys001
[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
[

(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-topics.sh --create --
topic quickstart-events --bootstrap-server localhost:9092
Created topic quickstart-events.
[(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-producer.sh ]
--topic quickstart-events --bootstrap-server localhost:9092
>event 1
>
```

● ○ ● kafka\_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:MaxGCPa...

```
Last login: Tue Dec 17 11:50:10 on ttys002
[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
[

(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh --topi]
c quickstart-events --from-beginning --bootstrap-server localhost:9092
event 1
[]
```

● ○ ● kafka\_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:MaxGCPa...

```
Last login: Tue Dec 17 11:52:27 on ttys003
[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
[

(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh --topi]
c quickstart-events --from-beginning --bootstrap-server localhost:9092
event 1
[]
```





## Библиотеки для работы с Kafka на Python:

1. `confluent-kafka-python` разработана Confluent, компанией, стоящей за коммерческим развитием Kafka. Библиотека обеспечивает высокую производительность и поддерживает последние функции Kafka, включая управление потреблением сообщений.

Она написана на C и Python

2. `kafka-python` полностью написана на питоне, что делает ее легко устанавливаемой и используемой. Идеально подходит для разработчиков, которые предпочитают работать исключительно в питон-экосистеме, не опираясь на внешние зависимости С-библиотек. Следует учесть, что она может быть менее производительной по сравнению с `confluent-kafka-python`.

<https://kafka-python.readthedocs.io>

# kafka-python

kafka [2.4, 2.3, 2.2, 2.1, 2.0, 1.1, 1.0, 0.11, 0.10, 0.9, 0.8]

python [2.7 | 3.4 | 3.5 | 3.6 | 3.7 | 3.8]

coverage [86%]

build [unknown] license [Apache 2]

ibility

Python client for the Apache Kafka distributed stream processing system. kafka-python is designed to function much like the official java client, with a sprinkling of pythonic interfaces (e.g., consumer iterators).

og

kafka-python is best used with newer brokers (0.9+), but is backwards-compatible with older versions (to 0.8.0). Some features will only be enabled on newer brokers. For example, fully coordinated consumer groups – i.e., dynamic partition assignment to multiple consumers in the same group – requires use of 0.9 kafka brokers. Supporting this feature for earlier broker releases would require writing and maintaining custom leadership election and membership / health check code (perhaps using zookeeper or consul). For older brokers, you can achieve something similar by manually assigning different partitions to each consumer instance with config management tools like chef, ansible, etc. This approach will work fine, though it does not support rebalancing on failures. See [Compatibility](#) for more details.

Please note that the master branch may contain unreleased features. For release documentation, please see [readthedocs](#) and/or python's inline help.

```
>>> pip install kafka-python
```

## KafkaConsumer

Installed ▾

Channels

Update index...

kafka X

Name	T	Description	Version
kafka-python	✓	O	2.0.2

---

```
[2]: from kafka import KafkaProducer  
from kafka.errors import KafkaError
```

```
[4]: producer = KafkaProducer(bootstrap_servers=['192.168.1.144:9092'])
```

```
[6]: future = producer.send('quickstart-events', value=b'raw_bytes_148')
```

```
[ ]:
```

✖️ ✎ ↑ ↓ ± ⌂ ✎



```
[2]: from kafka import KafkaProducer  
from kafka.errors import KafkaError
```

```
[4]: producer = KafkaProducer(bootstrap_servers=['192.168.1.144:9092'])
```

```
[6]: future = producer.send('quickstart-events', value=b'raw_bytes_148')
```

```
[8]: future = producer.send('quickstart-events', value=b'raw_bytes_149')
```

```
[ ]:
```



```
ding or offsets and group metadata from __consumer_offsets & for epoch 0 (kafka.coordinator.group.GroupCoordinator)
```

```
inputer for partition 25 in epoch 0 (kafka.coordinator.group.GroupCoordinator)
```

```
d kafka_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:Max
```

```
i>Last login: Tue Dec 17 11:52:27 on ttys003
```

```
n(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
```

```
n(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh -
```

```
n-topic quickstart-events --from-beginning --bootstrap-server localhost:9092
```

```
n event 1
```

```
n event 2
```

```
n raw_bytes_148
```

```
n raw_bytes_149
```

```
n
```

## kafka.ipynb

```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           auto_offset_reset='earliest',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')`  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                           message.offset, message.key,  
                                           message.value.decode('utf-8')))
```

```
quickstart-events:0:0: key=None value=event 1  
quickstart-events:0:1: key=None value=event 2  
quickstart-events:0:2: key=None value=raw_bytes_148  
quickstart-events:0:3: key=None value=raw_bytes_149
```

[ ]:



```
kafka_2.13-3.9.0 ━ java -Xmx512M -server -XX:+UseG1GC -XX:Ma...
Last login: Tue Dec 17 11:38:54 on ttys001
(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server localhost:9092
Created topic quickstart-events.
(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server localhost:9092
>event 1
>event 2
>event 3
>
offsets and group metadata from __consumer_offsets-2 in 11 milliseconds for
```

```
ka [kafka_2.13-3.9.0] - java -Xmx512M -server -XX:+UseG1GC -X

a. Last login: Tue Dec 17 11:50:10 on ttys002
e[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
e[
ep (base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consu
e opic quickstart-events --from-beginning --bootstrap-server localhost:9092
e event 1
e event 2
e raw_bytes_148
e raw_bytes_149
ep event 3
e □
ep
e
e
e
le
e
e
e
e
e
ep
e
e
e
e
e
ep
e
```

## kafka.ipynb

```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           auto_offset_reset='earliest',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')'  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                            message.offset, message.key,  
                                            message.value.decode('utf-8')))
```

```
quickstart-events:0:0: key=None value=event 1  
quickstart-events:0:1: key=None value=event 2  
quickstart-events:0:2: key=None value=raw_bytes_148  
quickstart-events:0:3: key=None value=raw_bytes_149  
quickstart-events:0:4: key=None value=event 3
```

[ ]:



## Kafka-Copy1.ipynb

```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')`  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                           message.offset, message.key,  
                                           message.value.decode('utf-8')))
```

[ ]:



```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')`  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                           message.offset, message.key,  
                                           message.value.decode('utf-8')))
```

quickstart-events:0:5: key=None value=raw\_bytes\_150

```
inputer for partition 0 in epoch 0 (kafka-coordinator-group-GroupCoordinator)
d: ● ● ● kafka_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:Max...
i:
d: Last login: Tue Dec 17 11:52:27 on ttys003
n:[base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
n:[base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh -
n:-topic quickstart-events --from-beginning --bootstrap-server localhost:9092
n: event 1
n: event 2
n: raw_bytes_148
n: raw_bytes_149
n: event 3
n: raw_bytes_150
n: □
n:
n:
n:
n:
n:
p:
n:
```

```
[ka] [ ] [ ] kafka_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -  
[a. Last login: Tue Dec 17 11:50:10 on ttys002  
[e[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0  
[e[  
[ep (base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-con-  
[e opic quickstart-events --from-beginning --bootstrap-server localhost:  
[e event 1  
[e event 2  
[e raw_bytes_148  
[e raw_bytes_149  
[ep event 3  
[e raw_bytes_150  
[ep [ ]  
[e  
[e  
[e  
[e  
[le  
[e  
[e  
[e  
[e  
[e  
[e  
[e  
[ep  
[e  
[e
```

```
laptop:~ sepo$ kafka_2.13-3.9.0 -- java -Xmx512M -server -XX:+UseG1GC -XX:Max...
d: Last login: Tue Dec 17 11:52:27 on ttys003
(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0
(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh --
n -topic quickstart-events --from-beginning --bootstrap-server localhost:9092
n event 1
n event 2
n raw_bytes_148
n raw_bytes_149
n event 3
n raw_bytes_150
n event 4
n
n
n
n
n
n
n
n
n
n
n
n
```

```
ka [● ● ●] kafka_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -  
a. Last login: Tue Dec 17 11:50:10 on ttys002  
e[(base) sepo@MacBookPro16 ~ % cd kafka_2.13-3.9.0  
e[  
ep (base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-cons  
e opic quickstart-events --from-beginning --bootstrap-server localhost:  
e event 1  
e event 2  
e raw_bytes_148  
e raw_bytes_149  
ep event 3  
e raw_bytes_150  
ep event 4  
e [ ]  
e  
le  
e  
e  
e  
e  
e  
ep  
e  
e
```

## kafka.ipynb

```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')`  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                         message.offset, message.key,  
                                         message.value.decode('utf-8')))  
  
quickstart-events:0:5: key=None value=raw_bytes_150  
quickstart-events:0:6: key=None value=event 4
```

## Kafka-Copy1.ipynb

```
[2]: from kafka import KafkaConsumer  
  
[4]: consumer = KafkaConsumer('quickstart-events',  
                           auto_offset_reset='earliest',  
                           bootstrap_servers=['localhost:9092'])  
  
[*]: for message in consumer:  
    # message value and key are raw bytes -- decode if necessary!  
    # e.g., for unicode: `message.value.decode('utf-8')`  
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                         message.offset, message.key,  
                                         message.value.decode('utf-8')))  
  
quickstart-events:0:0: key=None value=event 1  
quickstart-events:0:1: key=None value=event 2  
quickstart-events:0:2: key=None value=raw_bytes_148  
quickstart-events:0:3: key=None value=raw_bytes_149  
quickstart-events:0:4: key=None value=event 3  
quickstart-events:0:5: key=None value=raw_bytes_150  
quickstart-events:0:6: key=None value=event 4
```

[ ]:



- **group\_id** (str or None) – The name of the consumer group to join for dynamic partition assignment (if enabled), and to use for fetching and committing offsets. If None, auto-partition assignment (via group coordinator) and offset commits are disabled. Default: None

Имя группы потребителей для присоединения к динамическому назначению разделов (если включено) и использования для извлечения и фиксации смещений. Если None, автоматическое назначение разделов (через координатора группы) и фиксации смещений отключены. По умолчанию: None

## Kafka-Copy2.ipynb

```
[4]: from kafka import KafkaConsumer

[6]: consumer = KafkaConsumer('quickstart-events',
                             group_id ='g_1',
                             auto_offset_reset='earliest',
                             bootstrap_servers=['localhost:9092'])

[*]: for message in consumer:
    # message value and key are raw bytes -- decode if necessary!
    # e.g., for unicode: `message.value.decode('utf-8')`
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,
                                           message.offset, message.key,
                                           message.value.decode('utf-8')))

quickstart-events:0:0: key=None value=event 1
quickstart-events:0:1: key=None value=event 2
quickstart-events:0:2: key=None value=raw_bytes_148
quickstart-events:0:3: key=None value=raw_bytes_149
quickstart-events:0:4: key=None value=event 3
quickstart-events:0:5: key=None value=raw_bytes_150
quickstart-events:0:6: key=None value=event 4
```

[ ]:





jupyter kafka-Copy2 Last Checkpoint: 5 minutes ago

File Edit View Run Kernel Settings Help

Trusted

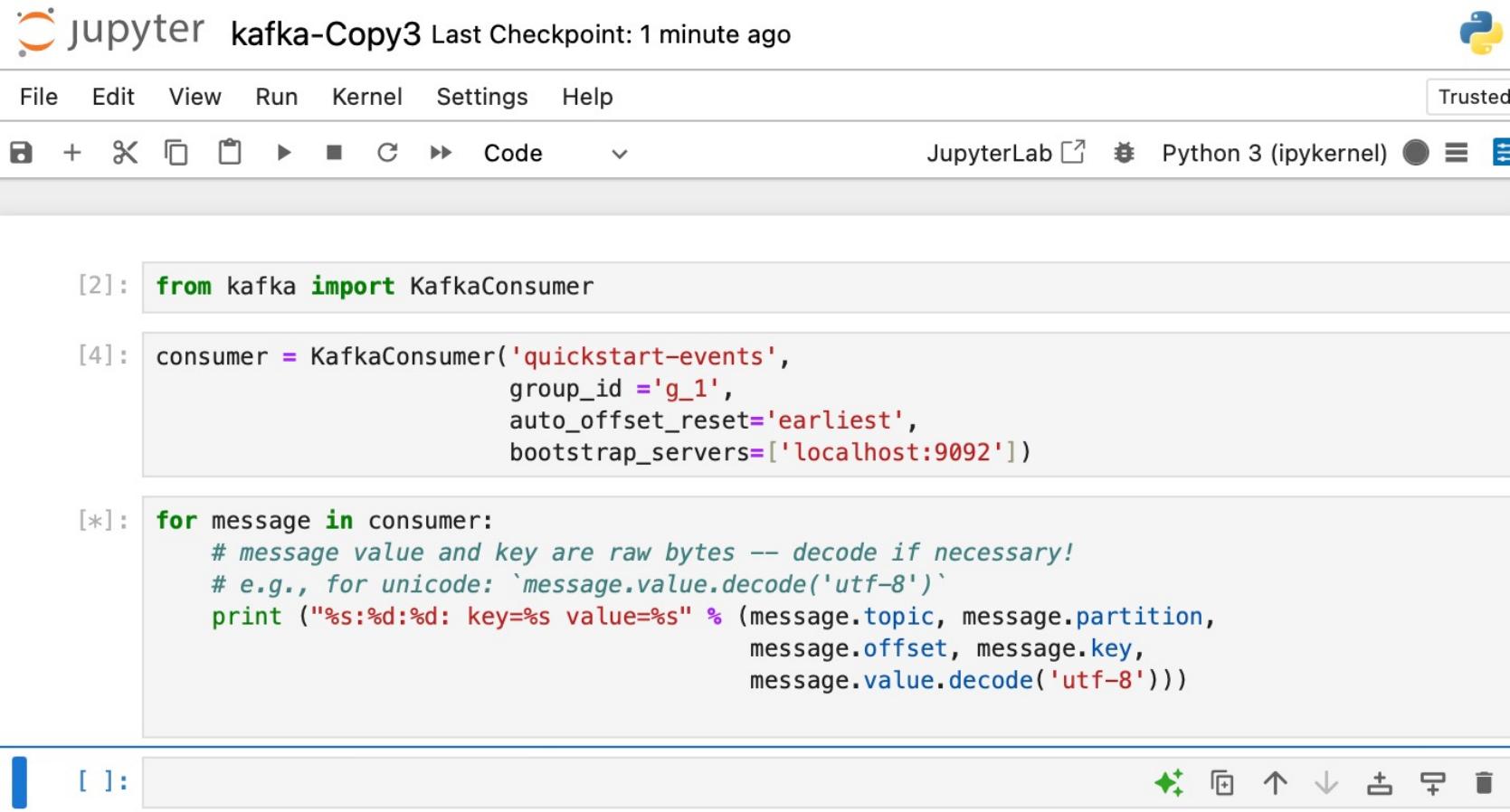
JupyterLab  Python 3 (ipykernel)  

JupyterLab  Python 3 (ipykernel)  

```
[4]: from kafka import KafkaConsumer
```

```
quickstart-events:0:0: key=None value=event 1
quickstart-events:0:1: key=None value=event 2
quickstart-events:0:2: key=None value=raw_bytes_148
quickstart-events:0:3: key=None value=raw_bytes_149
quickstart-events:0:4: key=None value=event 3
quickstart-events:0:5: key=None value=raw_bytes_150
quickstart-events:0:6: key=None value=event 4
```

[ ] :



Kafka-Copy3.ipynb

```
[2]: from kafka import KafkaProducer
from kafka.errors import KafkaError

[4]: producer = KafkaProducer(bootstrap_servers=['192.168.1.144:9092'])

[6]: future = producer.send('quickstart-events', value=b'raw_bytes_148')

[8]: future = producer.send('quickstart-events', value=b'raw_bytes_149')

[10]: future = producer.send('quickstart-events', value=b'raw_bytes_150')

[12]: future = producer.send('quickstart-events', value=b'raw_bytes_151')

[ ]:
```

```
[2]: from kafka import KafkaConsumer

[4]: consumer = KafkaConsumer('quickstart-events',
                            group_id ='g_1',
                            auto_offset_reset='earliest',
                            bootstrap_servers=['localhost:9092'])

[*]: for message in consumer:
    # message value and key are raw bytes -- decode if necessary!
    # e.g., for unicode: `message.value.decode('utf-8')`
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,
                                            message.offset, message.key,
                                            message.value.decode('utf-8')))

    quickstart-events:0:7: key=None value=raw_bytes_151
    quickstart-events:0:8: key=None value=raw_bytes_152
```

ng offsets and group me  
ni kafka\_2.13  
ni Last login: Tue Dec 1  
ni (base) sepo@MacBookPr  
ni (base) sepo@MacBookPr  
ni -topic quickstart-eve  
ni event 1  
ni event 2  
ni raw\_bytes\_148  
ni raw\_bytes\_149  
ni event 3  
ni raw\_bytes\_150  
ni event 4  
ni raw\_bytes\_151  
ni raw\_bytes\_152



**kafka** Last Checkpoint: last month



File Edit View Run Kernel Settings Help

Trusted

A set of small, light-gray navigation icons located at the bottom of the slide. From left to right, they include: a document icon, a plus sign, a scissor icon, a double arrow icon, a clipboard icon, a right-pointing arrow, a square icon, a 'C' icon, a double-right-pointing arrow icon, the word 'Code', and a downward-pointing arrow.

JupyterLab  Python 3 (ipykernel)    

```
[2]: from kafka import KafkaConsumer
```

```
quickstart-events:0:0: key=None value=event 1
quickstart-events:0:1: key=None value=event 2
quickstart-events:0:2: key=None value=raw_bytes_148
quickstart-events:0:3: key=None value=raw_bytes_149
quickstart-events:0:4: key=None value=event 3
quickstart-events:0:5: key=None value=raw_bytes_150
quickstart-events:0:6: key=None value=event 4
quickstart-events:0:7: key=None value=raw_bytes_151
quickstart-events:0:8: key=None value=raw_bytes_152
quickstart-events:0:9: key=None value=raw_bytes_153
```

[ ] :

## STEP 8: TERMINATE THE KAFKA ENVIRONMENT

Now that you reached the end of the quickstart, feel free to tear down the Kafka environment—or continue playing around.

1. Stop the producer and consumer clients with `Ctrl-C`, if you haven't done so already.
2. Stop the Kafka broker with `Ctrl-C`.
3. Lastly, if the Kafka with ZooKeeper section was followed, stop the ZooKeeper server with `Ctrl-C`.

If you also want to delete any data of your local Kafka environment including any events you have created along the way, run the command:

```
$ rm -rf /tmp/kafka-logs /tmp/zookeeper /tmp/kraft-combined-logs
```

```
[1]: from kafka import KafkaProducer  
from kafka.errors import KafkaError  
from kafka.structs import TopicPartition  
  
from kafka.admin import KafkaAdminClient, NewTopic
```

```
[3]: producer = KafkaProducer(bootstrap_servers=['localhost:9092'])
```

```
[ ]: future = producer.send('quickstart-events', value=b'raw_bytes_148')
```

- [11]: future = producer.send(topic, partition=0, value=b'0, 200, 210, 220')  
future = producer.send(topic, partition=1, value=b'1, 201, 211, 221')

```
[13]: producer.partitions_for('quickstart-events')
```

```
[13]: {0}
```

```
[ ]: producer.close()
```

Добавляем разделы в топик:

```
[ ]: from kafka import KafkaAdminClient
from kafka.admin import NewPartitions

topic = 'quickstart-events'
bootstrap_servers = 'localhost:9092'

admin_client = KafkaAdminClient(bootstrap_servers=bootstrap_servers)
topic_partitions = {}
topic_partitions[topic] = NewPartitions(total_count=3)
admin_client.create_partitions(topic_partitions)
```

Создаём новый топик:

```
[ ]: from kafka.admin import KafkaAdminClient, NewTopic

topic_list = []
topic_list.append(NewTopic(name='riders-events', num_partitions=2, replication_factor=1))
admin_client.create_topics(new_topics=topic_list, validate_only=False)

[13]: producer.partitions_for('quickstart-events')

[13]: {0, 1, 2}

[15]: topic1 = 'riders-events'
producer.partitions_for(topic1)

[15]: {0, 1}
```

```
[1]: from kafka import KafkaConsumer  
from kafka.structs import TopicPartition
```

```
[3]: consumer = KafkaConsumer('quickstart-events',  
                           group_id ='g_1',  
                           auto_offset_reset='earliest',  
                           consumer_timeout_ms=10000,  
                           bootstrap_servers=['localhost:9092'])
```

```
[17]: for message in consumer:  
      # message value and key are raw bytes -- decode if necessary!  
      # e.g., for unicode: `message.value.decode('utf-8')`  
      print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,  
                                              message.offset, message.key,  
                                              message.value.decode('utf-8')))
```

```
[19]: consumer.partitions_for_topic('quickstart-events')
```

```
[19]: {0, 1, 2}
```

```
[81]: producer.partitions_for('quickstart-events')
```

```
[81]: {0, 1, 2}
```

```
[75]: future = producer.send('quickstart-events', partition=0, value=b'raw_bytes_160')
future = producer.send('quickstart-events', partition=1, value=b'raw_bytes_161')
future = producer.send('quickstart-events', partition=2, value=b'raw_bytes_162')
```

## kafka-Copy2.ipynb

```
[2]: from kafka import KafkaConsumer
```

```
[4]: consumer = KafkaConsumer('quickstart-events',
                             group_id ='g_1',
                             auto_offset_reset='earliest',
                             bootstrap_servers=['localhost:9092'])
```

```
[*]: for message in consumer:
    # message value and key are raw bytes -- decode if necessary!
    # e.g., for unicode: `message.value.decode('utf-8')`
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,
                                           message.offset, message.key,
                                           message.value.decode('utf-8')))
```

```
quickstart-events:0:0: key=None value=raw_bytes_148
quickstart-events:0:1: key=None value=raw_bytes_149
quickstart-events:0:2: key=None value=raw_bytes_150
quickstart-events:0:3: key=None value=raw_bytes_151
quickstart-events:0:4: key=None value=raw_bytes_152
quickstart-events:0:5: key=None value=raw_bytes_153
quickstart-events:1:0: key=None value=raw_bytes_154
quickstart-events:1:1: key=None value=raw_bytes_155
quickstart-events:2:0: key=None value=raw_bytes_162
```

```
[17]: consumer.assignment()
```

```
[17]: {TopicPartition(topic='quickstart-events', partition=2)}
```

## kafka-Copy3.ipynb

```
[1]: from kafka import KafkaConsumer
```

```
[3]: consumer = KafkaConsumer('quickstart-events',
                             group_id ='g_1',
                             auto_offset_reset='earliest',
                             bootstrap_servers=['localhost:9092'])
```

```
[*]: for message in consumer:
    # message value and key are raw bytes -- decode if necessary!
    # e.g., for unicode: `message.value.decode('utf-8')`
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,
                                           message.offset, message.key,
                                           message.value.decode('utf-8')))
```

```
quickstart-events:0:6: key=None value=raw_bytes_156
quickstart-events:0:7: key=None value=raw_bytes_157
quickstart-events:1:2: key=None value=raw_bytes_161
quickstart-events:0:8: key=None value=raw_bytes_160
```

```
[21]: consumer.assignment()
```

```
[21]: {TopicPartition(topic='quickstart-events', partition=0),
       TopicPartition(topic='quickstart-events', partition=1)}
```

```
● ● ● kafka_2.13-3.9.0 — java -Xmx512M -server -XX:+UseG1GC -XX:MaxGCPause...
quickstart-events --bootstrap-server localhost:9092
[Topic: quickstart-events      TopicId: piTypbKUQjGB6J13_pvHRg PartitionCount: 1]
  ReplicationFactor: 1  Configs: segment.bytes=1073741824
    Topic: quickstart-events      Partition: 0      Leader: 1      Replicas: 1
  Isr: 1  Elr:  LastKnownElr:
(base) sepo@MacBookPro16 kafka_2.13-3.9.0 % bin/kafka-console-consumer.sh --topic qu
ickstart-events --from-beginning --bootstrap-server localhost:9092
raw_bytes_148
raw_bytes_149
raw_bytes_150
raw_bytes_151
raw_bytes_152
raw_bytes_153
raw_bytes_148
raw_bytes_149
raw_bytes_150
raw_bytes_151
raw_bytes_152
raw_bytes_153
raw_bytes_154
raw_bytes_155
raw_bytes_154
raw_bytes_155
raw_bytes_148
raw_bytes_149
raw_bytes_150
raw_bytes_151
raw_bytes_152
raw_bytes_153
raw_bytes_156
raw_bytes_157
raw_bytes_162
raw_bytes_161
raw_bytes_160
```

```
[93]: admin_client.list_consumer_groups()
```

```
[93]: [('console-consumer-72749', 'consumer'), ('g_1', 'consumer')]
```

## kafka-Copy3.ipynb

```
[19]: consumer.partitions_for_topic('quickstart-events')
```

```
[19]: {0, 1, 2}
```

```
[25]: tps = consumer.assignment()  
tps
```

```
[25]: {TopicPartition(topic='quickstart-events', partition=0),  
       TopicPartition(topic='quickstart-events', partition=1)}
```

```
[23]: consumer.beginning_offsets(tps)
```

```
[23]: {TopicPartition(topic='quickstart-events', partition=0): 0,  
       TopicPartition(topic='quickstart-events', partition=1): 0}
```

```
[ ]: consumer.seek_to_beginning(*tps)
```

```
[ ]: consumer.position(*tps)
```

```
[15]: tps = consumer.assignment()
print(tps)
for p in tps:
    print(p, consumer.position(p))

{TopicPartition(topic='quickstart-events', partition=0), TopicPartition(topic='quickstart-events', partition=1)}
TopicPartition(topic='quickstart-events', partition=0) 0
TopicPartition(topic='quickstart-events', partition=1) 0
```

```
[13]: for p in tps:
    consumer.seek_to_beginning(p)
```

```
[21]: consumer.assignment()
```

```
[21]: {TopicPartition(topic='quickstart-events', partition=0),
      TopicPartition(topic='quickstart-events', partition=1)}
```

```
[35]: producer = KafkaProducer(bootstrap_servers=[bootstrap_servers],
                           key_serializer=lambda key: key.encode('utf-8'),
                           value_serializer=lambda value: value.encode('utf-8'))
```

```
[71]: consumer = KafkaConsumer(topic1,
                           auto_offset_reset='earliest',
                           consumer_timeout_ms=2000,
                           key_deserializer=lambda key: int(key.decode('utf-8')),
                           value_deserializer=lambda value: value.decode('utf-8'),
                           bootstrap_servers=[bootstrap_servers])
```

```
[1]: from kafka import KafkaConsumer
from kafka.structs import TopicPartition <-- Не забываем!
```