Kafka[illegible]

Spring Kafka

Kafka[illegible]

[illegible]

Kafka[illegible]

[illegible]Kafka[illegible]/[illegible]

[illegible]/[illegible]RabbitMQ[illegible]

[illegible]

[illegible]/[illegible]

[illegible]/[illegible]

image-20251023233518657

[illegible]

image-20251023233533002

[illegible]

image-20251023233546291

Kafka[illegible]

[illegible]

- `Producer`
- `Consumer`
- `Kakfa Cluster`

[illegible]

image-20251023233656992

- `Producer` [illegible] `Kafka broker` [illegible]
- `Consumer` [illegible] `Kafka broker` [illegible]
- `Consumer Group(CG)` [illegible] `consumer` [illegible]
  - [illegible]
  - [illegible]

image-20251025101503778

- `Broker` [illegible]**[illegible] `Kafka` [illegible] `broker` **[illegible] `broker` [illegible] `broker` [illegible] `topic`

- `Topic` □□□□□□□□□□□□□□□□□□□□□□□□□□ `topic`
- `Partition` □□□□□□□□□□□ `Topic` □□□□□□□ `broker` □□□□□□□ `Topic` □□□□□□ `Partition` □□□□□□□□□□□

image-20251025101859263

- `Replica` □□□□□□ `topic` □□□□□□□□□□□□□□□ `leader` □□□□ `follower`
- `Leader` □ `Partition` □□□□□□□□□
- `Follower` □ `Partition` □□□□□□□□□

Kafka API

□□□□□□□□Spring□□□□□□□Kafka

□□□□

- `KafkaTemplate` □□□□□□□
- `@KafkaListener` □□□□□□□

□□□□□□□□□□□□□□□□□□□□

- `ProducerFactory` □□□□□□□□
- `ConsumerFactory` □□□□□□□□□□□□□□□
- `ConcurrentKafkaListenerContainerFactory` □□□□□□□□□□□□□□□
- `KafkaProducer` □□□□Kafka□□□□□□□□ `KafkaTemplate` □□□□□□□□□□□□□□
- `DefaultKafkaProducerFactory` □□ `ProducerFactory` □□□□□□□□□□□□□□ `KafkaProducer` □□
- `ContainerFactory` □□□□□□□□□□□□□□□ `MessageListenerContainer` □□□□□□□□□□□□□

□□□

□□□□

- □□□□

```xml
<dependency>
  <groupId>org.springframework.kafka</groupId>
  <artifactId>spring-kafka</artifactId>
  <version>3.3.10</version>
</dependency>
```

- □□□□□
  - □yaml□□□□□□□□□□

```yaml
spring:
  kafka:
    bootstrap-servers: localhost:9092  # Kafka□□□□□□□□□□□□□
    client-id: my-app                  # □□□□□□□□□□□□□□□□□□
    properties:                        # □□□□□Kafka□□□□□□□□□
      security.protocol: PLAINTEXT     # □□□□□PLAINTEXT□SASL_PLAINTEXT□□
     producer:
      acks: all                        # □□□□□0(□□□□)□1(leader□□)□all(□□□□□□)
      retries: 3                       # □□□□□□□□□□
      batch-size: 16384                # □□□□□□□□
      buffer-memory: 33554432          # □□□□□□□□□□
      linger-ms: 10                    # □□□□□□□□□□□□
      compression-type: gzip           # □□□□□none/gzip/snappy/lz4/zstd
      enable-idempotence: true         # □□□□□□□□□□□□
      key-serializer: org.apache.kafka.common.serialization.StringSerializer
      value-serializer:
org.springframework.kafka.support.serializer.JsonSerializer
      transaction-id-prefix: tx-           # □□□□
      properties:                      # □□□Kafka□□□□
        max.request.size: 1048576      # □□□□□□□□□1MB□
```

- □□□□□□□□□ `Spring` □□□□□
  - □□□□ `DefaultKafkaProducerFactory`
    - □□□□□□□□□□□□□□□□** `KafkaProducer` □□**□□□□Kafka□□□□
    - `KafkaTemplate` □□□□□ **`KafkaProducer`** □□□
  - □□□□ `KafkaTemplate`
  - □□□□□□□□□□ `KafkaTemplate` □□□

- □□□□□□□□□□□□□□□□□□□
  - □□ `KafkaTemplate` □□□□□□□□□□□□□□□□□□□□□□□□□

```java
@Service
public class KafkaProducerService {

    private final KafkaTemplate<String, String> kafkaTemplate;

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

    public void send(String topic, String msg) {
        kafkaTemplate.send(topic, msg);
        System.out.println("✅ □□□□: " + msg);
    }
}
```

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□ `KafkaTemplate` Bean
  - □□□□□□□□□□□□□□□□□□□□
  - □□□□□ `ProducerFactory` □□□□□□□□□□□□□□□□□□□□□□□

```java
@Configuration
public class KafkaProducerConfig {

    @Bean
    public KafkaTemplate<String, String> kafkaTemplate(
            ProducerFactory<String, String> producerFactory) {
```

```java
        return new KafkaTemplate<>(producerFactory);
    }
}
```

□□□□□□□□□

- □□□□

```java
// send(String topic, Integer partition, K key, V value)

kafkaTemplate.send("test-topic", 0, "myKey", "Hello Partition 0!");
```

- □□□□□□
  - □□ `MessageBuilder` □□□□
    - `withPayload` □□□□□□□
    - `KafkaHeaders.TOPIC` □□□□□□□□□
    - `KafkaHeaders.PARTITION` □□□□□□□□□
    - `KafkaHeaders.KEY` □□□□Key
    - `KafkaHeaders.TIMESTAMP` □□□□□□□□□□

```java
kafkaTemplate.send(
    MessageBuilder.withPayload("Hello Partition & Header")
        .setHeader(KafkaHeaders.TOPIC, "test-topic")
        .setHeader(KafkaHeaders.PARTITION_ID, 1)    // □□□□
        .setHeader(KafkaHeaders.MESSAGE_KEY, "myKey") // □□ key
        .setHeader("traceId", "trace-001")          // □□□ header
        .build()
);
```

□□□□

`SpringKafka` □□□□□□□□

- □□□□□ `CompletableFuture` □□□□□□□□□ `whenComplete` □□□□□□□□

```java
kafkaTemplate.send("topic", "key", "message")
    .whenComplete((result, ex) -> {
        if (ex != null) {
            System.err.println("□□□□: " + ex.getMessage());
        } else {
            System.out.println("□□□□, offset=" +
result.getRecordMetadata().offset());
        }
    });
```

□□-□□□□

□□ `ReplyingKafkaTemplate` □□□□□□ □□□□□□□□□□□□□□ Kafka □□□ RPC□□□□

- □□**topic**

- Spring□□□□□□□□□□□□□ `ReplyingKafkaTemplate`

```
@Configuration
public class KafkaReplyConfig {

    // □□□□□
    @Bean
    public ProducerFactory<String, String> producerFactory() {
        Map<String, Object> props = new HashMap<>();
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
StringSerializer.class);
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
StringSerializer.class);
        return new DefaultKafkaProducerFactory<>(props);
    }

    // □□□□□□□□□□□□□
    @Bean
    public ConsumerFactory<String, String> consumerFactory() {
        Map<String, Object> props = new HashMap<>();
        props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
        props.put(ConsumerConfig.GROUP_ID_CONFIG, "reply-group");
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class);
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
StringDeserializer.class);
        return new DefaultKafkaConsumerFactory<>(props);
    }

    // □□ "reply-topic" □□□□ReplyingKafkaTemplate □□□
    @Bean
    public KafkaMessageListenerContainer<String, String> replyContainer(
            ConsumerFactory<String, String> cf) {

        ContainerProperties containerProperties = new ContainerProperties("reply-
topic");
        return new KafkaMessageListenerContainer<>(cf, containerProperties);
    }

    // ReplyingKafkaTemplate —— □□□□
    @Bean
    public ReplyingKafkaTemplate<String, String, String> replyingKafkaTemplate(
            ProducerFactory<String, String> pf,
            KafkaMessageListenerContainer<String, String> replyContainer) {
        return new ReplyingKafkaTemplate<>(pf, replyContainer);
    }
}
```

- □□□□□□□□□□□

```java
@Service
public class RequestProducer {

    @Autowired
    private ReplyingKafkaTemplate<String, String, String> replyingKafkaTemplate;

    public String sendAndReceive(String data) throws Exception {
        // □□□□□□
        ProducerRecord<String, String> record =
                new ProducerRecord<>("request-topic", data);

        // □□□□□□□□Reply-To□
        record.headers().add(new RecordHeader(
                KafkaHeaders.REPLY_TOPIC, "reply-topic".getBytes()));

        // □□□□□□□□□□□□
        RequestReplyFuture<String, String, String> future =
                replyingKafkaTemplate.sendAndReceive(record);

        // □□□□□□□□□□ reply □□□
        ConsumerRecord<String, String> response = future.get(10, TimeUnit.SECONDS);

        System.out.println("□□□□□" + response.value());
        return response.value();
    }
}
```

□□□

□□□□

```
@KafkaListener
    ↓ (□□□□□)
ContainerFactory (□□)
    ↓ (□□□□□)
MessageListenerContainer (□□)
    ↓ (□□□□□□)
KafkaConsumer (□□□□□)
    ↓ (□□□□)
Kafka Broker
```

□□□□

- □□□□

```xml
<dependency>
  <groupId>org.springframework.kafka</groupId>
  <artifactId>spring-kafka</artifactId>
  <version>3.3.10</version>
</dependency>
```

- □□□□□□

```yaml
spring:
  kafka:
    bootstrap-servers: localhost:9092  # Kafka□□□□□□□□□□□□□
    client-id: my-app                  # □□□□□□□□□□□□□□□□□□
    properties:                        # □□□□□Kafka□□□□□□□□□
      security.protocol: PLAINTEXT     # □□□□□PLAINTEXT□SASL_PLAINTEXT□□
    consumer:
      group-id: my-group               # □□□□ID
      auto-offset-reset: earliest      # □□□□□□□□□□□□□□
      enable-auto-commit: true         # □□□□□□offset
      auto-commit-interval: 1000       # □□□□□□□□□□
      max-poll-records: 500            # □□□□□□□□□□□
      fetch-min-size: 1                # □□□□□□□□□□
      fetch-max-wait: 500              # □□□□□□□□□□□□□□(ms)
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      value-deserializer: 
org.springframework.kafka.support.serializer.JsonDeserializer
      properties:
        spring.json.trusted.packages: "*"   # □□□□□□□□□□□□
```

- □□□□□□□□□□Spring□□□□
  - □□□□ **`ConsumerFactory`**
  - □□□□ **`ConcurrentKafkaListenerContainerFactory`**
  - □□□□□□ Bean □ `@KafkaListener` □□□□□□□
- □□ `@KafkaListener` □□□□□□□□□□
  - □□□ `factory` □□□ **`spring.kafka.consumer.\*`** □□

```java
@KafkaListener(topics = "test-topic")
public void listen(String message) {
    System.out.println("□□□□: " + message);
}
```

- □□□ `Header` □□□
  - □ `@Header` □□□□□□□□□□□□□□□□
    - `KafkaHeaders.RECEIVED_TOPIC`
    - `KafkaHeaders.RECEIVED_PARTITION`
    - `KafkaHeaders.OFFSET`
    - `KafkaHeaders.RECEIVED_TIMESTAMP`
    - □□□ header□□□ traceId□

```java
@KafkaListener(topics = "orders")
public void listenWithKey(@Header(KafkaHeaders.RECEIVED_KEY) String key,
                          @Payload String value) {
    System.out.println("key=" + key + ", value=" + value);
}
```

□□□□□□□

□□□□□ `ContainerFactory`

- □□□□□□
- □□□□□□
- □□

□□

- □□□□□□□□□

```java
@Bean
public ConcurrentKafkaListenerContainerFactory<String, String>
kafkaListenerContainerFactory() {
    ConcurrentKafkaListenerContainerFactory<String, String> factory =
            new ConcurrentKafkaListenerContainerFactory<>();
    factory.setConsumerFactory(consumerFactory()); // □□□ ConsumerFactory
    factory.setConcurrency(3); // □□□□□
    factory.setBatchListener(true); // □□□□
    return factory;
}
```

- □□□ `@KakfaListener` □□

```java
@KafkaListener(topics = "test-topic", containerFactory =
"kafkaListenerContainerFactory")
public void listen(String message) { ... }
```

□□□□□□□

Kafka□□□□□□□□□□□ `__consumer_offsets` □□□□□□□□□□ `offset` □□□□□□□□□

- □□□□□□

```yaml
spring:
 kafka:
  consumer:
      enable-auto-commit: false # □□□□□□
```

- □□□□□□□ `ContainerFactory`
  - □□□ `AckMode` □
    - `RECORD` □□□□□□□□□□□□□
    - `BATCH` □□□□□□□□□□□□
    - `TIME` □□□□□
    - `COUNT` □□□□□□□□□□□□□
    - `MANUAL` □□□□□□□ `ack.acknowledge()`
    - `MANUAL_IMMEDIATE` □□□□□offset

```java
@Configuration
public class KafkaManualAckConfig {

    @Bean
    public ConcurrentKafkaListenerContainerFactory<String, String> manualFactory(
            ConsumerFactory<String, String> consumerFactory) {

        ConcurrentKafkaListenerContainerFactory<String, String> factory =
                new ConcurrentKafkaListenerContainerFactory<>();
        factory.setConsumerFactory(consumerFactory);

        // □□□□□□□□

factory.getContainerProperties().setAckMode(ContainerProperties.AckMode.MANUAL_IMMEDIA

        return factory;
    }
}
```

- □ `@KafkaListener` □□ `Acknowledgement` □□□□□□□□

```java
 @KafkaListener(topics = "manual-topic", containerFactory = "manualFactory")
    public void consume(ConsumerRecord<String, String> record, Acknowledgment ack) {
        try {
            System.out.println("□□□□: " + record.value());
            // □□□□
            ack.acknowledge();
            System.out.println("□□□□□□□□ offset");

        } catch (Exception e) {

            System.err.println("□□□□□□: " + record.value());
        }
    }
```

□□-□□□□

□□□□□□□□□□□□□□□□□□□□□□

- □ `@KafkaListener` □□□□□□□□ `@SendTo` □□□□□□□□□□□□□Topic

```java
@Component
public class ReplyConsumer {

    // □□□□□□□□□□□□□□□□□ reply-topic
    @KafkaListener(topics = "request-topic", groupId = "reply-group")
    @SendTo("reply-topic")  // □□□□□□□□□□□
    public String handleRequest(String message) {
        System.out.println("□□□□□" + message);
        return "□□□□□" + message.toUpperCase();
    }
}
```

□□□□□□□

□Kafka□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□offset

```yaml
spring:
 kafka:
  consumer:
      enable-auto-commit: false # □□□□□□
```

- □□ `DefaultErrorHandler` □□□□□□□
  - □□ **`DeadLetterPublishingRecoverer`** □□□□□□
    - □□ `KafkaTemplate` □□□□□□□□□□□□□□□
  - □□ **`FixedBackOff`** □□□□□□□□□
  - □□ **`DefaultErrorHandler`** □□□□□□□
    - □□ `DeadLetterPublishingRecoverer` □ `FixedBackOff`

```java
@Configuration
public class KafkaConsumerConfig {

    @Bean
    public DefaultErrorHandler errorHandler(KafkaTemplate<Object, Object> template)
{

        // □□ DLT□□□□□□□□□□
        DeadLetterPublishingRecoverer recoverer = new DeadLetterPublishingRecoverer(
            template,
            (record, ex) -> new TopicPartition("main-topic-dlt", record.partition())
        );

        // □□□□□□□□□□□□□□ 2s□□□□□ 3 □□
        FixedBackOff backOff = new FixedBackOff(2000L, 3);

        //  □□□□□□□
        DefaultErrorHandler errorHandler = new DefaultErrorHandler(recoverer,
backOff);

        // □□□□□□□□□□□□□□□□□
        errorHandler.addNotRetryableExceptions(IllegalArgumentException.class);

        // □□□□□□□□
        errorHandler.setRetryListeners((record, ex, deliveryAttempt) ->
            System.err.printf("□□□ %d □□□□%s%n", deliveryAttempt, ex.getMessage())
        );

        return errorHandler;
    }
}
```

- □□□□□ `@RetryableTopic` □□□□□□□□□□□□

```java
@RetryableTopic(
    attempts = "4", // □□□□ 4 □
    backoff = @Backoff(delay = 2000), // □□□□ 2s
    dltTopicSuffix = "-dlt" // □□□□□□
)
@KafkaListener(topics = "main-topic", groupId = "test-group")
public void consume(String message) {
    System.out.println("□□□□□" + message);
    throw new RuntimeException("□□□□");
}
```

Kafka Connect

□□

- □□□□□
- □□curl□□□□□□□□

Kafka Streams

□□□□

```xml
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-stream-binder-kafka-streams</artifactId>
</dependency>
```

□□□□□

- `spring.cloud.stream.function.definition` □□□ `Spring Cloud Stream` □□□□□□□Bean
  - □□□□□□□bean□□□□□ `;` □□□□
- `spring.cloud.stream.function.bindings` □□□□□□□□□□□□□□□□□□**Topic**□□□□□□□□□□
  - □□□□□□□□□□□□
    - `{function-name}-in-{index}`
    - `{function-name}-out-{index}`
  - `destination` □Kafka□Topic□□
- `spring.cloud.stream.kafka.streams.binder` □
  - `application-id` □Kafka Streams□□□□□ID
  - `brokers` □Kafka□□□□
  - `configuration` □Kafka Streams □□□□□□
    - `commit.interval.ms` □□□□□□□□ `changelog topic` □□□
    - `cache.max.bytes.buffering` □□□□□□□ RocksDB □□□□
    - `state.dir` □□□□□□□□

        - `processing.guarantee` □□□□□
            - `at_least_once` □□□□□
            - `exactly_once_v2` □□□□□
        - `default.key.serde` □□□ key □□□□□
        - `default.value.serde` □□□ value □□□□□
        - `num.stream.threads` □Kafka Streams □□□□□

- `spring.cloud.stream.binders` □□□□□□□□□Kafka□□□□□□□

```yaml
spring:
  cloud:
    stream:
      function:
        definition: process
      bindings:
        process-in-0:
          destination: input-topic
        process-out-0:
          destination: output-topic
      kafka:
        streams:
          binder:
            application-id: uppercase-app
            brokers: localhost:9092
            configuration:
              commit.interval.ms: 1000
              cache.max.bytes.buffering: 10485760
```

□□□□□Bean

- □□□□□□□□□□□□□□□□Bean□□
- □□□□□□□□□□ `Function<KStream<String, String>, KStream<String, String>>`
- [□□KStream API]()

```java
@Configuration
public class KafkaStreamsConfig {

    @Bean
    public Function<KStream<String, String>, KStream<String, String>> process() {
        return input -> input
                .filter((key, value) -> value.contains("hello"))
                .mapValues(String::toUpperCase);
    }
}
```

□□□□□□□□□

> □□□□□□□□□□□□□□□□□□□□

Kafka□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Redis□□□□□□□□

- □□ `Topic` □□□□□□□ `Partition`
- □□ `Partition` □□□□□□□□□□□□□□□□□□
    - □□□□□□ **Segment** □□□□
        - `.log` □□□□□
        - `.index` □□□□ → □□□□
        - `.timeindex` □□□□ → □□□

```
/kafka-logs/
 ├── topicA-0/
 │    ├── 00000000000000000000.log
 │    ├── 00000000000000000000.index
 │    └── 00000000000000000000.timeindex
 ├── topicA-1/
 │    ├── ...
```

Kakfa□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□600MB/s
- `Page Cache` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□Linux□ **`sendfile()`** □□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□/□□□□□□
- Kafka□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    - Producer □□□□
    - Broker □□□□□
    - Consumer □□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    - `.index` □□□□ → □□□□
    - `.timeindex` □□□□ → □□□

KRaft

Kafka□□□□□□□□□□□□□□□**Raft**□□□□ZooKeeper

- **RAFT** □□
    - □□□□
    - □□□□
    - □□□
- □KRaft□□□□□□□□□□ `Broker` □□□□□ `Controller`
    - Controller□□□□□□□
        - `Controller Leader` □□□□□□□□□□□□□□□□□□□□□□**Leader**□□□**Topic**□□□**Partition**□
        - `Controller Follower` □□□□□□□□□

□□□□

- □Kafka Broker□□□□□□□ `server.properties` □□□□□□□□□□□□□□□□□□□□□□□□□□□
    - ** `process.roles` **□□□□□□□□□□
    - ** `node.id` **□□□□□□□□□id
    - ** `listeners` **□□□□□□□□

        - ** `PLAINTTEXT` **□□□□□□□□□□□□/□□□□□□□□□□□□
        - ** `CONTROLLER` **□□□ `Controller` □□□□□□□□□□□□
    - ** `metadata.log.dir` **□□□□□□□□
    - **`controller.quorum.voters`** □**Raft**□□□□□□□□□□□□□□□ **`Controller leader`**
    - ** `log.dirs` **□□□□□□□
    - ** `num.partitions` **□□□□□topic□□□□□
    - ** `default.replication.factor` **□□topic□□□□□

```
# -------------------------
# □□□□
# -------------------------
process.roles=broker,controller   # □□□ broker □ controller
node.id=1                         # □□□□ ID

# -------------------------
# □□□□
# -------------------------
listeners=PLAINTEXT://localhost:9092,CONTROLLER://localhost:9093

# -------------------------
# □□□□□□□
# -------------------------
metadata.log.dir=/tmp/kraft-metadata-logs

# -------------------------
# Controller Raft quorum□□□□□□□□□□
# -------------------------
controller.quorum.voters=1@localhost:9093

# -------------------------
# □□□□□□
# -------------------------
log.dirs=/tmp/kafka-logs

# -------------------------
# □□ topic □□
# -------------------------
num.partitions=1
default.replication.factor=1
```

□□□□

- □□□□□□ `Broker` □□**controller quorum**
- □□ Raft □□□□□□□□□ `Controller Leader`
    - □□□□□□□□□(majority vote□
- □□□ `Controller Leader` □□□
    - □□/□□ topic
    - □□□□□□
    - Leader/Follower □□
    - ISR □□□□□

- □□□□□□ Follower□□ Leader □□□□□□□
- □□ Controller Leader □□□
  - □□□□□□□□□□□□□□□□□□□□□□□□□□□ Leader
  - □ Leader □□□□□□□□□□□□□

□□

□□□□□□□

- □□□□
- □□□□
- □□□□□□□□
- □□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

`Kafka` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

> ***Producer* □ + *Broker* □ + *Consumer* □ □□□□□□□□**

Kafka□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□
  - □□□□□□□□ `acks = 0` □□
  - □□□□□□□ `offset`
- □□□□□□□□□□□□□□□□□□□□□□□□□
  - □□□□□□ `acks=1` □ `acks=all` □□□□□□□
  - □□□□□□□□□□ `offset`
- □□□□□□□□□□□□□□□□□□□□□□□□
  - □□□□□□□ **+** □□□□□□
  - Kafka□□□□□□□□□□ **`Producer ID`** □□□□□□□□□□□□□ **`sequence number`**
  - □□□□□□□□□□□□□□□□□□ `PID + sequence number` □□□□□□ `offset` □□□□□□□□□□□□□□□□□□
    - □□□□□□ `offset` □□□□□□□
    - □□□□□□** `offset` □□□□□□□**
  - `Broker` □□
  - □□□□□ `broker` □□□□□□□□□□□□□□□□□□□□□□□□□

□□

Kafka□□□ `Topic` □□□□□□ `Partition` □□□□□□□□□□ `broker` □□□□Replication□

- □□□□□□□ **Leader**□□□□□□□□□ **Follower**□□□□□
- □□□□□□□□□□□□□□□□**Leader**□□
- Follower□□Leader□□□□□□□□□□□□□□□
- Kafka □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- □□**ISR**□□□□□□□□□□□□□□Producer□□□□
- □□□□□□□□□

□□□**ISR**□□□□□□□□□□□□□□

- □□ISR□□□□□□□□□□
  - □□□□□□□□□□□□□□□□□
- □□□□□□□□□□Leader□□□□□□□
  - □□□□□□□□□□□□

□□□□□**ISR**□

Kafka□□□□□□□□**ISR**□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

- Broker□□□□□□
- □□□Follower□□□□□□□□□□□□□□□□
  - **Follower**□□□□□□□□□**ISR**□□□□□□□□□□□□□□□

□Leader□□□□□□**Leader**□□□**ISR**□□□

- □□□□□□□□□□ `Controller` □□□□□□□□□□□□**Leader**□□
- `Controller` □□□ `Leader` □□
- □ISR□□□□□□□□□□□ `Leader` □□□□□□□
- □□□ `Leader` □□□□□Broker□ `Producer/Consumer`

□□□

□□□□□□□□□□□□□□ `broker` □□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□□□□
- □□□□□□□□□□key□□□□□□□□□□□key□□□□□□□

□□□□□

- □□□□□□□□□□□□□□□□□□□□
  - □□□□□□□ `batch.size`
  - □□□□ `linger.ms`
  - □□□□□□□□□
- □□□□□□□□□I/O□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□

- **Acks 0**□□□□□□□□□□□□□□□□□□□□□□□Kafka□□
  - □□□□□□□□□□□□□□□□□□
- **Acks 1**□□□□□□□□ `Leader` □□□□□□□
  - □□□□□□□□□□□□□□□
- **Acks -1**□□□□□□□□** `Leader` □□□ `follower` **□□□□□□□
  - □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

Producer□□□□□

- **Range**□□ key □□□□□□□□□□□□

- **RoundRobin**□□□□□□□
- **Sticky**□□□□□□□□□□□□□□□□□□□□□□□□□□ batch □□□□□□□□□□□□□

□□□

□□□□

- □□□□□□ `broker` □□□□
- □□□□□□□□□□□
- □□□□□□□□□□□□
- □□□□□□□□□□□□□□□□

□□□□

- `broker` □□□□□□□□□□
- □□□□□□□□□□□

□□□□

- □□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□ `group.id` □□□□□□

□□□□□□

- □□□□□□□□□□□□□□□□□□□□□□
- Kafka□□□□□□□□□ `Topic` □ `__consumer_offsets` □□□□□□□□□□□□□□□□□□□□□□□□□□□

image-20251026155003645

□□□□□□□□4□□□□

- □□□□□□□6□□□
- □□□□□□□□1□□□□□□□□□□
- □□□□□□□□□□□□□□□□□10□□
- □□□□□□□□□□□□□□□□14
- □□□□□□□□□□□**$\le$**□□□□□□□□□□□□□□

□□□□□Compact□

- Kafka□□□□□□□□□□□□□□□Topic□ `__consumer_offsets` □□□□□□□□□□□□□□□□□offset
  - `key = consumer group + partition`
  - `value = □□□offset`
- Kafka□ `__consumer_offsets` □□ □□□□
  - □□□□ key□□□□ `consumer group + partition` □□□□□□□□□□□
  - □□□□□□□□offset

□□□□

Consumer□□□□□□

- Kafka□□□□□□□□□□□□□□□□□□□□□□
  - **Range**□□□□□□□□□□□□□

- **RoundRobin**[illegible]
- **Sticky**[illegible]

image-20251026163046158

[illegible]**Rebalance**[illegible]

[illegible]

- [illegible]
  - [illegible]
    - [illegible]`offset`
    - [illegible]`JoinGroup`[illegible]`Group Coordinator`
    - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]`offset`[illegible]
  - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]
- [illegible]
  - [illegible]
    - [illegible] `JoinGroup` [illegible]
    - [illegible]
    - [illegible]
    - [illegible]
    - [illegible]
  - [illegible]

Kafka Connect

Kafka Connect[illegible]Kafka[illegible]Kafka[illegible]`Elasticsearch`[illegible]`Hadoop`[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]

[illegible]

- ** `Connectors` **[illegible]
- ** `Tasks` **[illegible]`Connectors`[illegible]
- ** `Workers` **[illegible] `Connector` [illegible] `Task` [illegible]
- ** `Conveters` **[illegible]/[illegible]
- ** `Transforms` **[illegible]`Connector`[illegible]
- ** `Dead Letter Queue` **[illegible]`Connector`[illegible]

[illegible]Connector[illegible]

□□

- ** `Source Connector` **□□□□□□□□□□□□Kafka
- ** `Sink Connector` **□□Kafka□□□□□□□□□□

□□□**Task**□

`Task` □ `Connector` □□□□□□□□□□□□□□□□□□□□□□Kafka□□□Kafka□□□□□□

□□□□□

- □□□□□□□□□□□□□
- □□□□□□□□□□ Kafka □□□□□□Topic□□□□□□ `Task` □□
- □□ `Connector` □□□□□ `Task`
- `Task` □□□□□□□□□□□□□□
- `Task` □□□□□□□□□□□□□□□□□□□□□□□□□□□□ REST API □□

□□

- **`Standalone`** □□□□□□□□□ `Worker` □□□□□□ `Connector/Task`
- **`Distributed`** □□□□□□□□□□ `Worker` □□□□□□□□□□□□□□

□□□**Worker**□

`Worker` □□□ `Connector` □ `Task` □□□

□□□□□□□□□□

- □□□□□**Standalone**□□□□□□□□□□□
- □□□□□□**Distributed**□□□□Worker□□□□□□□□□□□□□□□□□□□□□□

□□□□**Converter**□

□□□□□□□□□□□□□□□□□□/□□□□

- □□□□□□□□□□□ `Kafka` □□□□□□
- □ `Kafka` □□□□□□□□ `Kafka` □□□□□□□□□□□□
- □□□□□□□□□□ Connector □□□□□□ Transform□

□□□□□□**Transform**□

□□□□□□□□□□□□□□

- □□□□□□
- □□□□□□□□□
- □□ `Topic` □

□□□□□□□□□□□□□□□

□□□□□**Dead Letter Queue**□

□ `Task` □□□□□□□□□□□□□□□□□□□□□□□□

□□

- □□□□□□□□□□□□□□□□□□□□□□□□□**DLQ**□□
  - **DLQ**□□□**Dead Letter Queue Topic**□□□□□□□□□□□□□□□□□□□□□□
- `errors.tolerance` □□□
  - `none` □□□□□□□□□□□□□□□□□□□□

    - `all` [illegible]DLQ

Kafka Streams

Kafka Streams[illegible] `Java` [illegible]Kafka[illegible]

[illegible]

- [illegible] `Topic` [illegible]
- [illegible]
- [illegible] `Topic`

[illegible]

Stream[illegible]

[illegible]

- [illegible]
- stream[illegible]Topic[illegible]

KStream

- [illegible]Stream[illegible]Kafka Streams[illegible]
- [illegible]record[illegible] `key[illegible]value[illegible]timestamp`
- `Key` [illegible]
    - [illegible]**Key**[illegible]Topic[illegible]
    - [illegible] `groupByKey()` [illegible] `join()` [illegible]Key

KTable[illegible]

[illegible] key [illegible]

- [illegible]key[illegible]
- [illegible]KTable[illegible]
- [illegible] `KTable` [illegible] Key [illegible]
- [illegible] `KTable` [illegible]**Kafka Topic**[illegible] `Changelog Topic`
    - [illegible]

GlobalKTable[illegible]

[illegible]

- [illegible]Kafka Streams[illegible] `Kafka Topic` [illegible] `GlobalKTable`
- [illegible] **`GlobalKTable`** [illegible]
- [illegible] `Kafka Topic` [illegible] `GlobalKTable` [illegible]
- [illegible]

Topology[illegible]

Kafka Streams [illegible]**DAG**[illegible]

- `Kafka Streams` [illegible]
    - [illegible]**source**[illegible]
    - [illegible]**processor**[illegible]
    - [illegible]**sink**[illegible]

- □Streams□□□□□□□□□□□□□□□□□□□□□

DSL□Domain Specific Language□

Kafka Streams □□□**API**□□□□□□□□□□□□□□□□□□□□□□

□□□□□

- □□ `Topology`
- □□ `state store`
- □□ `changelog`
- □□□□□□□□

State Store□□□□□□ changelog□□□□□

`Kafka Streams` □□□□□□□□□□

- □□□□□□□□ RocksDB □ □□□□□□□□□
- □ `KTable` □□□**Kafka Topic**□□ `Changelog Topic`
- □□□□□□□□□□ `changelog topic` □□□ state

Processor API

`Kafka Streams` □□□□**API**□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□ `Processor` □
- □□□□□□□ `Topology.addProcessor` □
- □□□□□□□ `StateStore` □

Window□□□

□□□□□□□□□□□□□□

- □□□□□□□□□□□
- □□□□□□□□□□
  - **Tumbling**□□□□□□□□□□□□□□□ 1 □□□□□
  - **Hopping**□□□□□□□□□□□□□□□□□□□□□□□ 1 □□□ 30s □□□□□
  - **Session**□□□□□□□□□□□□□□□ `gap` □□□□
- □□□□□□□□□□ `state` □ `WindowStore` □
- □□□□□□□ `retention` □□□□□□□□
- □□□□□ `event-time` □□□□□ `processing-time` □□□□□□□ `event-time + watermark` □□□

Watermark□GracePeriod

Watermark□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□

- □Watermark□□□□□□□□□□□□□□□□□□□□□□

- **Watermark**□□□□□□□□□□□□□□□□ □□□□□□□

- **Watermark**□□□□□□□□□□□□□□□□□□□

GracePeriod□□□□

- □□□

- [illegible]

[illegible]Processing Guarantees[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible]

Kafka Streams[illegible]commit[illegible]

[illegible]Time Semantics[illegible]

Kafka Streams [illegible]

- `Event Time` [illegible]
- `Processing Time` [illegible] Kafka Streams [illegible]
- `Ingestion Time` [illegible] Kafka broker [illegible]