

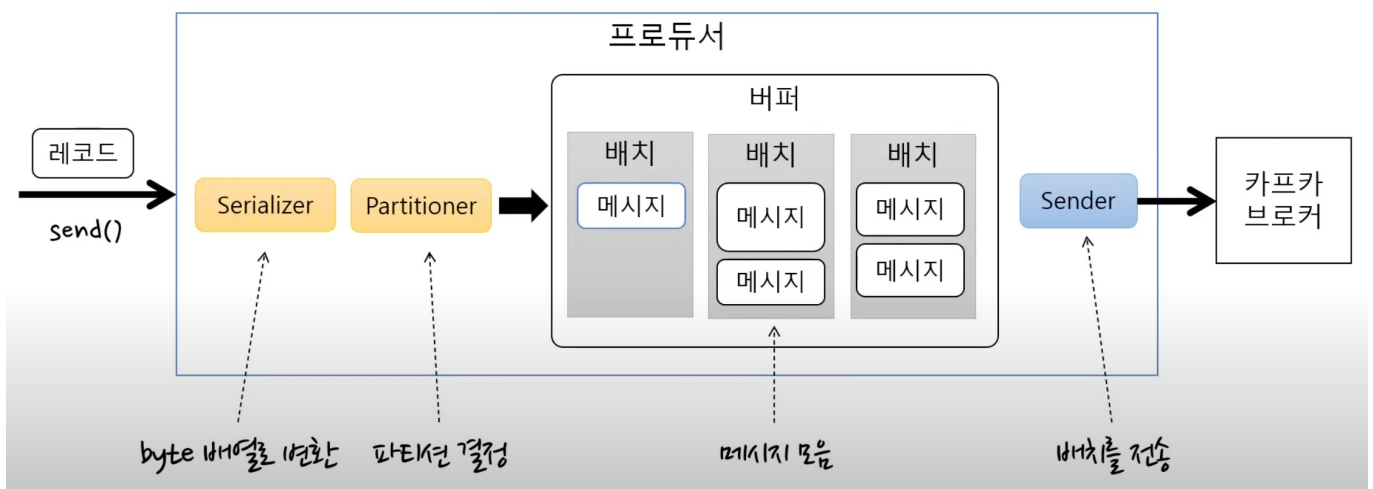
- ○ , ,

```
Properties prop = new Properties();
prop.put("bootstrap.servers", "kafka01:9092,kafka01:9092,kafka01:9092");
prop.put("key.serializer",
"org.apache.kafka.common.serialization.StringSerializer");
prop.put("value.serializer",
"org.apache.kafka.common.serialization.StringSerializer");

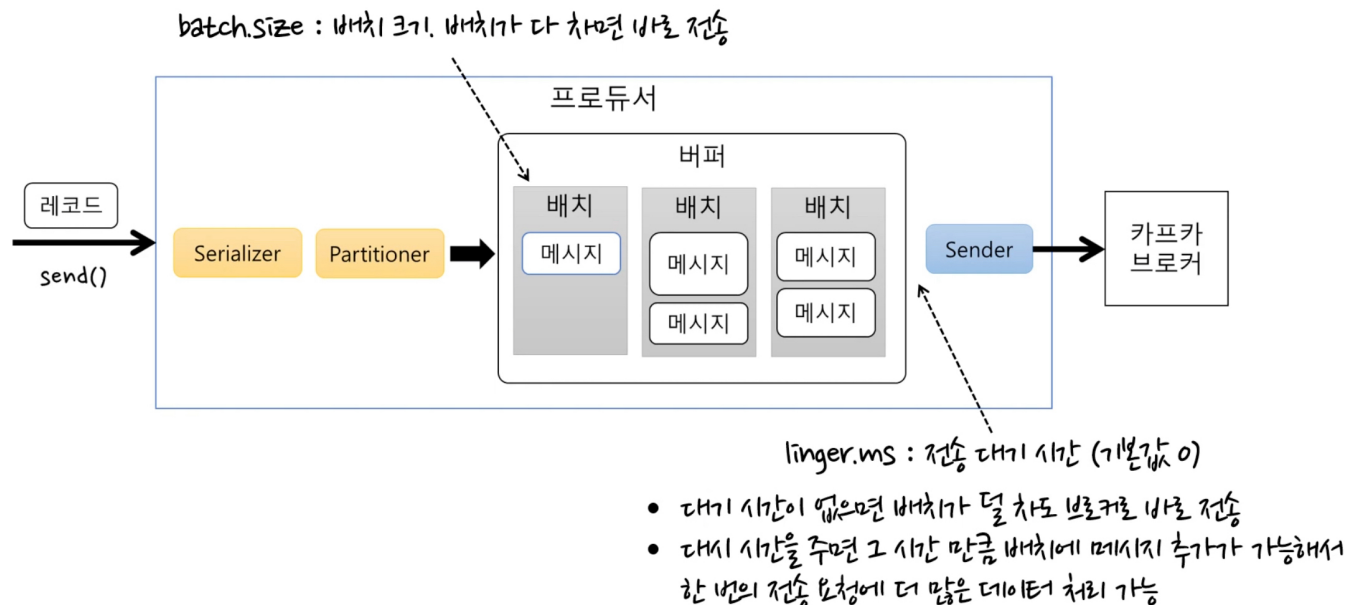
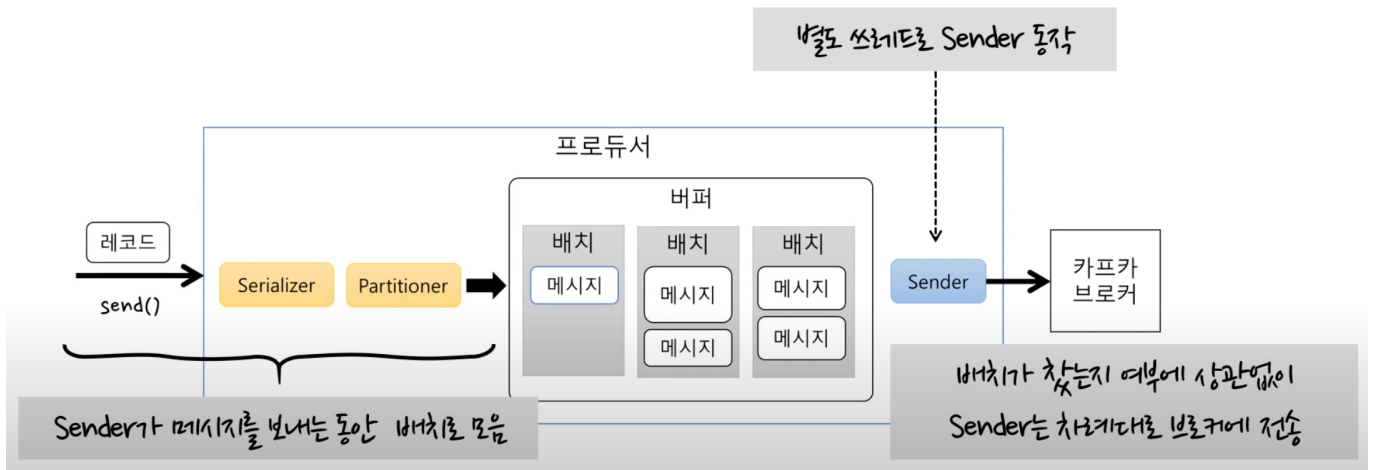
KafkaProducer<Integer, String> producer = new KafkaProducer<>(prop);

producer.send(new ProducerRecord<>("topicname", "key", "value"));
producer.send(new ProducerRecord<>("topicname", "value"));

producer.close();
```



Sender



```
producer.send(new ProducerRecord<>("simple", "value"));
```

•
•

가

: Future

```
Future<RecordMetadata> f = producer.send(new ProducerRecord<>("topic", "value"));
```

```
try {
```

```
RecordMetadata meta = f.get(); //
} catch (ExecutionException ex) {
}
}
```

- →
-

: Callback

```
producer.send(new ProducerRecord<>("simple", "value"),
    new Callback() {
        @Override
        public void onCompletion(RecordMetadata metadata, Exception ex) {
        }
    });
```

-

ack

- ack = 0
 -
 - zero
- ack = 1
 -
 - 가
- ack = all (-1)
 - - min.insync.replicas

ack + min.insync.replicas

- min.insync.replicas ()
 - ack all
- 1:
 - 3, ack = all, min.insync.replicas = 2
 -
- 2:
 - 3, ack = all, min.insync.replicas = 1
 -
 - ack = 1 (가)
- 3:
 - 3, ack = all, min.insync.replicas = 3
 - 2

- 가
- - ()
 -
 -
 -
- - , 가
 -
 -
- 1 :
- - 가
 - :
- - 가 가
 - retries
 - send() send()
 - send()
- 가
- 2 :
- - , DB
 - ()
- - send()
 - send()
 - send() 가 Future get()
- 가
- 가
- : enable.idempotence
- max.in.flight.requests.per.connection
 -
 - 1 가
 - 1 가

Refs

- <https://www.youtube.com/watch?v=geMtm17ofPY>

- [Kafka](#)

From:

<https://jace.link/> - **Various Ways**

Permanent link:

<https://jace.link/open/%ED%94%84%EB%A1%9C%EB%93%80%EC%84%9C>

Last update: **2021/11/08 01:41**

