

Consumer

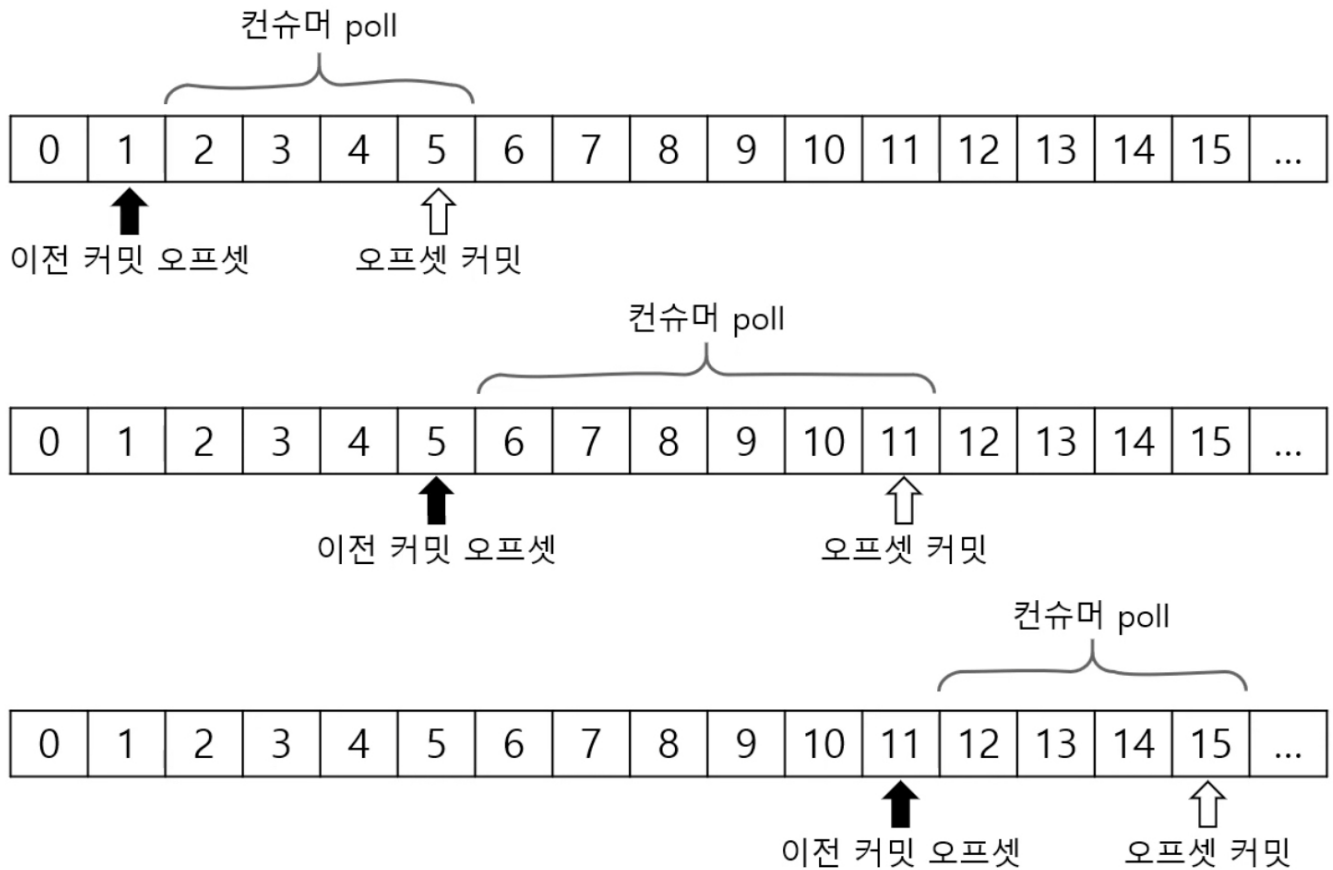
-

```
Properties prop = new Properties();
prop.put("bootstrap.servers", "localhost:9092");
prop.put("group.id", "group1");
prop.put("key.deserializer",
"org.apache.kafka.common.serialization.StringDeserializer");
prop.put("value.deserializer",
"org.apache.kafka.common.serialization.StringDeserializer");

KafkaConsumer<String, String> consumer = new KafkaConsumer<String,
String>(prop);
consumer.subscribe(Collections.singleton("simple")); //

while( ){
    ConsumerRecords<String, String> records =
consumer.poll(Duration.ofMillis(100));
    for (ConsumerRecord<String, String> record: records) {
        System.out.println(record.value() + ":" + record.topic() + ":" +
record.partition() + ":" + record.offset());
    }
}

consumer.close();
```



-
- `auto.offset.reset`
 - `earliest` :
 - `latest` : 가 ()
 - `none` :
- - `fetch.min.bytes`: 가
 - : 1
 - 가
 - `fetch.max.wait.ms`: 가 가 가
 - : 500
 - 가 poll()
 - `max.partition.fetch.bytes`: 가
 - : 1048576 (1MB)
- /
- `enable.auto.commit`
 - `true`: 가 ()
 - `false`:

- auto.commit.interval.ms:
 - : 5000 (5)
- poll(), close()

: /

```
ConsumerRecords<String, String> records =
consumer.poll(Duration.ofSeconds(1));
for (ConsumerRecord<String, String> record: records) {
...
}
try {
    consumer.commitSync();
} catch (Exception ex) {
    //
}
```

```
ConsumerRecords<String, String> records =
consumer.poll(Duration.ofSeconds(1));
for (ConsumerRecord<String, String> record: records) {
...
}
consumer.commitAsync(); // commitAsync(OffsetCommitCallback callback)
```

- 가
 - ,
- (idempotence)
 - :
 - 1 가 → 1 가 → 1 가
 - 2가 4가

- , **poll**

- 가
 - - session.timeout.ms : (10)
 - heartbeat.interval.ms : (3)
 - session.timeout.ms 1/3
- max.poll.interval.ms : poll()
 - poll()

- - poll() wakeup() 가 WakeupException → close()

```
KafkaConsumer<String, String> consumer = new KafkaConsumer<String,
String>(prop);
consumer.subscribe(Collections.singleton("simple"));

try {
    while (true) {
        ConsumerRecords<String, String> records =
consumer.poll(Duration.ofSeconds(1)); // wakeup()
        ... records
        try {
            consumer.commitAsync();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
} catch (Exception ex) {
    ...
} finally {
    consumer.close();
}
```

:

- KafkaConsumer
 - !
 - wakeup()

Refs

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

- [Kafka](#)

From:

<https://moro.kr/> - **Various Ways**

Permanent link:

<https://moro.kr/open/consumer>

Last update: **2022/01/09 12:02**

