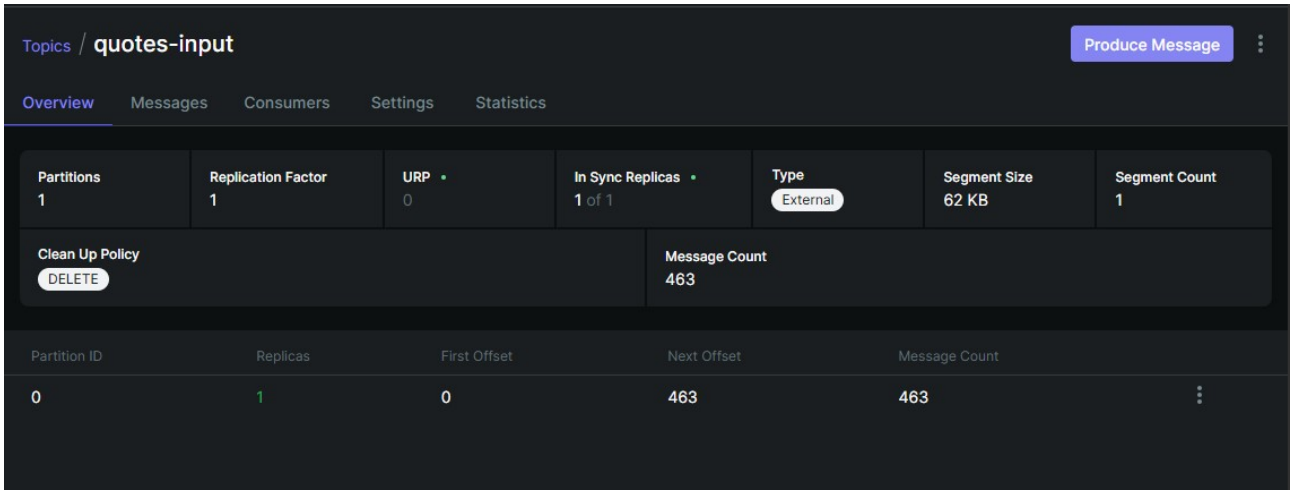
Exercise_03_basic_producer_consumer

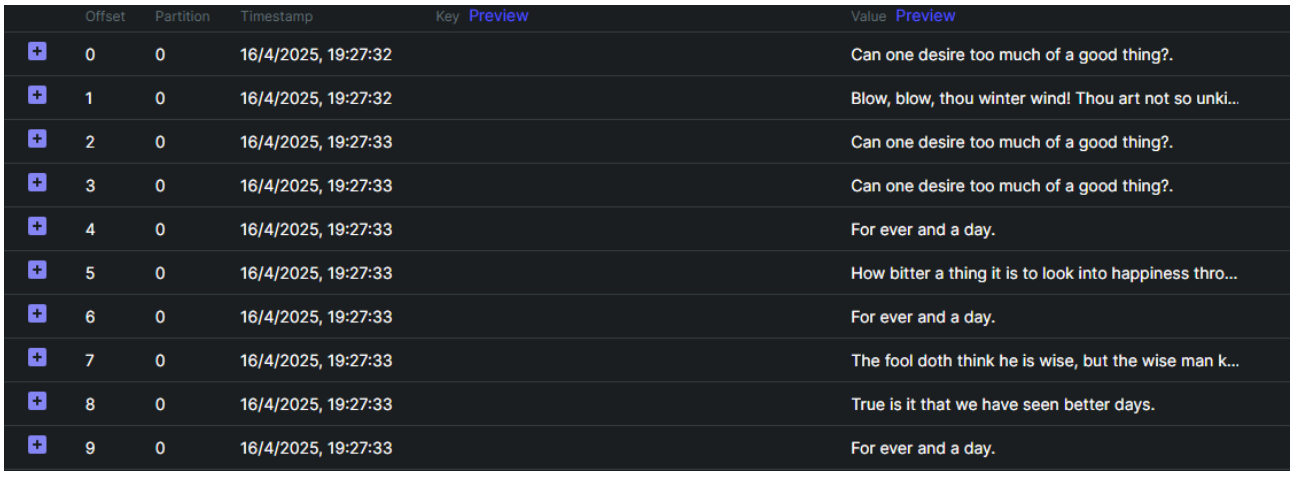
La entrega de la práctica debe ser en formato zip donde debe incluir:

- **Funcionalidad de Exercise 03 en formato pdf.**
- **Ficheros de API Java con las modificaciones.**

1.- Escribe el nombre del Topic donde el Producer está enviando los mensajes en la API Java



2.- Describe los mensajes qué esta enviando el Producer en la API Java.



3.- Describe qué hace el código de WordCountConsumer

```
static void createWordCountStream(final StreamsBuilder builder) {
    final KStream<String, String> textLines = builder.stream(INPUT_TOPIC);
    final Pattern pattern = Pattern.compile("\\W+", Pattern.UNICODE_CHARACTER_CLASS);
    final KTable<String, Long> wordCounts = textLines
            .map(((key, value) -> {
                System.out.println (" VALUE " + value);
                return KeyValue.pair(key, value);
            }))
            .flatMapValues(value -> Arrays.asList(pattern.split(value.toLowerCase())))
            .groupBy((keyIgnored, word) -> word)
            .count();

    wordCounts.toStream().to(OUTPUT_TOPIC, Produced.with(Serdes.String(), Serdes.Long()));

}
```

Este método permite crear una tabla de key value donde key es el identificador del contador de palabras y el value el número que cuenta cuantas palabras tiene la frase.

4.- Explica cómo el Consumer obtiene los mensajes según el código de la API Java.

```
try{
    while (true){
        ConsumerRecords<String, String> records = kafkaConsumer.poll(Duration.ofMillis(100));
        for (ConsumerRecord<String, String> record: records){
            System.out.println(" | " + record.key() + " | " +  String.valueOf(record.value()));
        }
    }
}catch (Exception e){
    System.out.println(e.getMessage());
}finally {
    kafkaConsumer.close();
}
```

Definimos un ConsumerRecords y recorremos sus records en formato string donde imprimimos el key value del hashmap de tipo string, string del ConsumerRecords.

5.-Realiza un pequeño cambio en el código de la API Java de Producer modificando el mensaje que se envía.

```
ttry{
    for(int i = 0; i < 100000; i++){
        Thread.sleep(100);
        Esports es = faker.esports();
        System.out.println("Event: " + es.event() + "  Game: " + es.game()  + "  League " + es.league()  + " Team  "  +
es.team()+ " Player  "  + es.player());
        String quote = faker.yoda().quote();
        System.out.println(quote);
        kafkaProducer.send(new ProducerRecord<String, String>("quotes-input", quote ));
    }
}catch (Exception e){
    e.printStackTrace();
}finally {
    kafkaProducer.close();
}
```