



Listado de consumidores de Kafka

Última modificación: 31 de agosto de 2020

por Mona Mohamadinia (<https://www.baeldung.com/author/mona-mohamadinia/>)
(<https://www.baeldung.com/author/mona-mohamadinia/>)

Nube (<https://www.baeldung.com/category/cloud/>)

Kafka (<https://www.baeldung.com/tag/kafka/>)

Acabo de anunciar el nuevo curso *Learn Spring*, centrado en los fundamentos de Spring 5 y Spring Boot 2:

>> VER EL CURSO (/ls-course-start)

1. Información general

En este tutorial rápido, aprenderemos cómo enumerar los grupos de consumidores de Kafka y también echaremos un vistazo a sus detalles.

2. Requisitos previos

Para ejecutar los ejemplos de este tutorial, necesitaremos un clúster de Kafka al que enviar nuestras solicitudes. Puede ser un clúster Kafka completo que se ejecuta en un entorno de producción, o puede ser un clúster Kafka de instancia única y ([/devops/kafka-list-topics#setting-up-kafka](#)) específico de prueba.

En aras de la simplicidad, asumiremos que tenemos un clúster de un solo nodo escuchando el puerto 9092 con una instancia de Zookeeper ([/java-zookeeper](#)) escuchando el puerto 2181 en el localhost.

Además, tenga en cuenta que estamos ejecutando todos los comandos de ejemplo desde el directorio de instalación de Kafka.

3. Agregar temas y consumidores

Antes de enumerar los consumidores en un clúster de Kafka en particular, agreguemos algunos temas (https://kafka.apache.org/documentation/#basic_ops_add_topic) primero usando el *script de shell* `kafka-topics.sh`:

```
1 $ ./bin/kafka-topics.sh --create --topic users.registrations --replication-factor 1 \  
2   --partitions 2 --zookeeper localhost:2181 \  
3 $ ./bin/kafka-topics.sh --create --topic users.verifications --replication-factor 1 \  
4   --partitions 2 --zookeeper localhost:2181
```

Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#)

Okay



Ahora, también necesitamos agregar algunos grupos de consumidores . La forma más sencilla es utilizar el consumidor de consola incluido en las distribuciones de Kafka :

```
1 $ ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic users.registrations --group new-user
2 $ ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic users.registrations --group new-user
```



Aquí, hemos utilizado el *script de shell* `kafka-console-consumer.sh` para agregar dos consumidores que escuchan el mismo tema. Estos consumidores están en el mismo grupo, por lo que los mensajes de las particiones de tema se distribuirán entre los miembros del grupo . De esta manera podemos implementar el patrón de consumidores competidores (<https://www.enterpriseintegrationpatterns.com/patterns/messaging/CompetingConsumers.html>) en Kafka.

Consumamos también de otro tema:

```
1 $ ./bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic users.verifications
```

Como no especificamos un grupo para el consumidor, el consumidor de la consola creó un nuevo grupo, con él mismo como el único miembro .

Veremos este nuevo grupo en la siguiente sección, donde aprenderemos cómo enumerar consumidores y grupos de consumidores en el clúster de Kafka.

4. Listado de consumidores

Para enumerar los consumidores en el clúster de Kafka, podemos usar el *script de shell* `kafka-consumer-groups.sh` . La opción `-list` enumerará todos los grupos de consumidores :

```
1 $ ./bin/kafka-consumer-groups.sh --list --bootstrap-server localhost:9092
2 new-user
3 console-consumer-40123
```

Además de la opción `-list` , estamos pasando la opción `-bootstrap-server` para especificar la dirección del clúster de Kafka. Tenemos tres consumidores individuales en dos grupos, por lo que el resultado contiene solo dos grupos.

Para ver los miembros del primer grupo , podemos usar las opciones `"-group <nombre> -describir -members"` :

```
1 $ ./bin/kafka-consumer-groups.sh --describe --group new-user --members --bootstrap-server localhost:9092
2 GROUP          CONSUMER-ID          HOST                CLIENT-ID          #PARTITIONS
3 new-user       consumer-new-user-1-b90... /127.0.0.1         consumer-new-user-1 1
4 new-user       consumer-new-user-1-af8... /127.0.0.1         consumer-new-user-1 1
```

Aquí, podemos ver que hay dos consumidores individuales en nuestro grupo de *nuevos usuarios* , cada uno consumiendo desde una partición.

Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#). Si omitimos la opción `-members` , enumerará los consumidores del grupo, el número de partición que cada uno está escuchando y sus compensaciones: Okay

```
1 $ ./bin/kafka-consumer-groups.sh --describe --group new-user --bootstrap-server localhost:9092
2 GROUP TOPIC PARTITION CURRENT-OFFSET LOG-END-OFFSET LAG
3 new-user users.registrations 1 3 3 0
4 new-user users.registrations 0 5 5 0
```



Una cosa más a tener en cuenta es que se requiere la dirección del servidor del clúster o de arranque para este comando. Si omitimos la información de conexión del clúster, el script de shell arrojará un error :

```
1 $ ./bin/kafka-consumer-groups.sh --list
2 Missing required argument "[bootstrap-server]"
3 // truncated
```

5. Conclusión

En este breve tutorial, agregamos algunos temas de Kafka y grupos de consumidores al principio. Luego, aprendimos cómo enumerar grupos de consumidores y ver los detalles de cada grupo.

Acabo de anunciar el nuevo curso *Learn Spring*, centrado en los fundamentos de Spring 5 y Spring Boot 2:

>> VER EL CURSO (/ls-course-end)



Cree su arquitectura de microservicio con
Spring Boot y Spring Cloud




Ingrese su dirección de correo electrónico








>> Descargar ahora

Iniciar sesión (https://www.baeldung.com/wp-login.php?redirect_to=https%3A%2F%2Fwww.baeldung.com%2Fdevops%2Flisting-kafka-consumers)



Be the First to Comment!

B *I* U    ” </>  {} [+]



0 COMENTARIOS



 **ezoic** (<https://www.ezoic.com/what-is-ezoic/>)
Quéjate de este anuncio

Usamos cookies para mejorar su experiencia con el sitio. Para obtener más información, puede leer la [Política de privacidad y cookies completa \(/privacy-policy\)](#)

CATEGORÍAS

Okay



[PRIMAVERA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/SPRING/\)](https://www.baeldung.com/category/spring/)
[DESCANSO \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/REST/\)](https://www.baeldung.com/category/rest/)
[JAVA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/\)](https://www.baeldung.com/category/java/)
[SEGURIDAD \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/\)](https://www.baeldung.com/category/security-2/)
[PERSISTENCIA \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/\)](https://www.baeldung.com/category/persistence/)
[JACKSON \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/JSON/JACKSON/\)](https://www.baeldung.com/category/json/jackson/)
[LADO DEL CLIENTE HTTP \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/HTTP/\)](https://www.baeldung.com/category/http/)
[KOTLIN \(HTTPS://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/\)](https://www.baeldung.com/category/kotlin/)

SERIE

[TUTORIAL DE JAVA 'VOLVER A LO BÁSICO' \(/JAVA-TUTORIAL\)](#)
[TUTORIAL DE JACKSON JSON \(/JACKSON\)](#)
[TUTORIAL DE HTTPCLIENT 4 \(/HTTPCLIENT-GUIDE\)](#)
[DESCANSO CON SPRING TUTORIAL \(/REST-WITH-SPRING-SERIES\)](#)
[TUTORIAL DE PERSISTENCIA DE PRIMAVERA \(/PERSISTENCE-WITH-SPRING-SERIES\)](#)
[SEGURIDAD CON SPRING \(/SECURITY-SPRING\)](#)

ACERCA DE

[SOBRE BAELDUNG \(/ABOUT\)](#)
[LOS CURSOS \(HTTPS://COURSES.BAELDUNG.COM\)](https://courses.baeldung.com)
[TRABAJOS \(/TAG/ACTIVE-JOB/\)](#)
[EL ARCHIVO COMPLETO \(/FULL_ARCHIVE\)](#)
[ESCRIBE PARA BAELDUNG \(/CONTRIBUTION-GUIDELINES\)](#)
[EDITORES \(/EDITORS\)](#)
[NUESTROS COMPAÑEROS \(/PARTNERS\)](#)
[ANUNCIE EN BAELDUNG \(/ADVERTISE\)](#)

[TÉRMINOS DE SERVICIO \(/TERMS-OF-SERVICE\)](#)
[POLÍTICA DE PRIVACIDAD \(/PRIVACY-POLICY\)](#)
[INFORMACIÓN DE LA COMPAÑÍA \(/BAELDUNG-COMPANY-INFO\)](#)
[CONTACTO \(/CONTACT\)](#)