



Kafka Consumer CLI Tutorial

How to consume data from Kafka?

The Kafka console consumer CLI, `kafka-console-consumer` is used to read data from Kafka and output it to standard output. Make sure you have [started Kafka](#) beforehand.

CLI Extensions

Use CLI commands with appropriate extensions for your platform, e.g., `kafka-console-consumer.bat` for windows, `kafka-console-consumer.sh` for Linux

How to Consume Data in a Kafka Topic using the CLI?

To create a Kafka topic, we need to provide the mandatory parameters:

Find your the Kafka hostname and port e.g., `localhost:9092`

Provide the mandatory parameters: topic name

If you need to read historical data, using the `--from-beginning` option.

Otherwise, you will only be reading future data

Use the `kafka-console-consumer.sh` CLI.

Important recommendation: do NOT use the `--zookeeper` option

Since Kafka v0.10, the consumer is leveraging a Kafka connection string, not Zookeeper. This is due to how consumer offsets are stored. Any tutorial you see using the `--zookeeper` option should be considered **outdated**.

Example

Make sure you have followed the [Kafka Producer CLI Tutorial](#) to create and send data to a Kafka topic first.

Consuming only the future messages of a Kafka topic

.sh

```
1 kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first_topic
```



Consuming all historical messages and future ones in a Kafka topic

Use the `--from-beginning` option

.sh

```
1 kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first_topic
```



To exit the Kafka console producer, use the keyboard combination `Ctrl+C`.

Command Output

```
1 Hello World
2 My name is Conduktor
3 I love Kafka
4 ^CProcessed a total of 3 messages (<-- when doing a Ctrl+C to exit)
```

The Kafka Console Consumer will remain opened until you exit it, and will keep on displaying messages to the screen. It assumes all the messages coming in can be deserialized as text (`String`).

By default, the Kafka Console Consumer does not show the key, or any partition information.

If you do not see any output but you know your Kafka topic has data in it, don't forget to use the `--from-beginning` option.

Important note on message ordering

The order of messages is not total, it is per partition. As a topic may be created with more than one partition, the order is only guaranteed at the partition level. If you try with only one partition, you will see total ordering.

Gotchas

Here are the common mistakes and caveats with the `kafka-console-consumer.sh` command:

Messages by default will not display the key or metadata information (see below for how to do it).

When you start a `kafka-console-consumer`, unless specifying the `--from-beginning` option, only future messages will be displayed and read.

If the topic does not exist, the console consumer will automatically create it with default

You can consume multiple topics at a time with a comma-delimited list or a pattern.

If a consumer group id is not specified, the `kafka-console-consumer` generates a random consumer group.

If messages do not appear in order, remember that the order is at the partition level, not at the topic level.

Extra Important options you can set (advanced)

`--from-beginning`

We won't repeat this one enough. To read all historical messages

`--formatter`

To display messages in a particular format (example below to display keys)

`--consumer-property`

To pass in any consumer property, such as the `allow.auto.create.topics` setting

`--group`

By default a random consumer group ID is chosen, but you can override it with this option. See the demo in the [Kafka Consumers in Group CLI Tutorial](#).

`--max-messages`

Number of messages to consume before exiting

`--partition`

If you want to only consume from a specific partition.

How to consume a Kafka Topic and show both the key and value using the Kafka Console Consumer CLI?

By default, the console consumer will show only the value of the Kafka record. Using this command you can show both the key and value.

Using the formatter `kafka.tools.DefaultMessageFormatter` and using the properties `print.timestamp=true` `print.key=true` `print.value=true`:

`.sh`

```
1 kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic first_topi
2
3 CreateTime:1641810588071      null      hello
4 CreateTime:1641823304170      name      Stephane
5 CreateTime:1641823301294      example key      example value
```

More properties are available such as:

`print.partition`

```
print.offset  
  
print.headers  
  
key.separator  
  
line.separator  
  
headers.separator
```

Was this content helpful? 

 2

 0

PREVIOUS

NEXT

Share this page.



Apache, Apache Kafka, Kafka and the Kafka logo are trademarks of the **Apache Software Foundation**. All other trademarks, servicemarks, and copyrights are the property of their respective owners.