

Project: Visualizing Live Streaming Data with Elixir, Kafka, and Chart.js

Storage Space

397 / 500MB (79.4%)

Active

Tasks >

Task 3: Produce Messages

<<

1

Use different processes that come with `os_mon` and generate data to be sent through the producer to Kafka.

✓

✓

In the `ProducerDemo` module, call the default `kaffe` method `produce_sync/2` to send messages to Kafka.

✓

Create a `push_memory/0` function in the `/usercode/producer/lib/producer_demo.ex` file that:

1. Sends a message containing memory used by processes using the `erlang.memory/1` method with the relevant arguments

2. Divides the result by `(1024*1024)` to convert it into MBs before sending it

3. Sends another message containing the percentage of CPU utilized using the `cpu_sup` process

4. Waits for two seconds

5. Calls itself again

2

Calling the `push_memory/0` function infinitely ensures that the messages are continuously sent to the Kafka server.

✓

3

Note: Convert the values returned from both methods to strings before sending them as messages.

If you need any help, click the “Show Hint” button below:

Hide Hint

• Use the `Kaffe.Producer.produce_sync(key, value)` function to send messages.

• Use the `:erlang.memory(:processes_used)` method to get memory usage.

• Use the `cpu_sup.utils()` method to get CPU utilization.

• Convert results to strings using the `to_string` method.

• Add a two-second pause using the `timer.sleep(2000)` method.

4

If you’re unsure how to produce messages, click the “Show Solution” button below:

Hide Solution

< Previous

✓ Completed

Next >

Search

Sea

/usercode

> .git

> TestElixir

> consumer

> kafka

> lost+found

> producer

Open a file by clicking from the file tree

?