Redis CLI experiments

How to Get Started

Try experimenting with the running code using the Redis CLI. To do so:

Run `python consumer_group.py` in a terminal window.

Start `redis-cli` another terminal window.

Now, run each of the following commands in the CLI.

Exercise #1

Get the length of the stream:

```
XLEN numbers
```

Running this command a number of times should show that the stream is constantly growing as the producer adds more numbers to it. For example:

```
127.0.0.1:6379> XLEN numbers

(integer) 148

127.0.0.1:6379> XLEN numbers

(integer) 159
```

Here we can see that between the two invocations of `XLEN`, the producer added another `11` messages to the stream.

Exercise #2

Get information about the stream:

```
XINFO STREAM numbers
```

The `XINFO STREAM` command shows information about the current state of the stream, including:

The stream's overall length

Information about the underlying radix tree implementation

The number of consumer groups associated with the screen (we have `1`, the **`primes`** group)

The last (highest) ID in the stream

The first and last entries in the stream

You should see output similar to the following:

```
127.0.0.1:6379> XINFO STREAM numbers
 1) "length"
 2) (integer) 489
 3) "radix-tree-keys"
 4) (integer) 5
 5) "radix-tree-nodes"
 6) (integer) 13
 7) "groups"
 8) (integer) 1
 9) "last-generated-id"
```

```
10) "1556600608383-0"
11) "first-entry"
12) 1) "1556600526564-0"
    2) 1) "n"
       2) "0"
13) "last-entry"
14) 1) "1556600608383-0"
    2) 1) "n"
       2) "488"
```

Exercise #3

Get information about consumer groups:

```
XINFO GROUPS numbers
```

The output from `XINFO GROUPS` shows us which consumer groups are associated with the stream. We have one group named `primes`, containing our three consumers instances:

```
127.0.0.1:6379> XINFO GROUPS numbers
1) 1) "name"
   2) "primes"
   3) "consumers"
   4) (integer) 3
   5) "pending"
   6) (integer) 48
   7) "last-delivered-id"
   8) "1556600638386-0"
```

The output from **`XINFO GROUPS`** command also shows us how many pending messages the group has (these messages have been delivered to the group's consumers but not yet acknowledged).

Exercise #4

Get information about consumers:

```
XINFO CONSUMERS numbers primes
```

The **XINFO CONSUMERS** command's output shows us information about the status of each consumer in the group **`primes`**:

```
127.0.0.1:6379> XINFO CONSUMERS
numbers primes
1) 1) "name"
   2) "BOB-0"
   3) "pending"
   4) (integer) 25
   5) "idle"
   6) (integer) 1276
2) 1) "name"
   2) "BOB-1"
   3) "pending"
   4) (integer) 24
   5) "idle"
   6) (integer) 1363
```

```
3) 1) "name"
   2) "BOB-2"
   3) "pending"
   4) (integer) 15
   5) "idle"
   6) (integer) 671
```

For each consumer in the group, we can see the consumer's name, how many pending messages it has, and how long it has been idle (milliseconds since it last acknowledged a message or read from the stream).

Exercise #5

Get information about the stream’s memory usage:

```
MEMORY USAGE numbers
```

This command will return the number of bytes used to store the stream's structure and values.

← Previous

Next →

Modules

Course overview

□ Lesson

Course overview

□ Lesson

Environment setup

Consumer groups

□ Lesson

The problem with slow consumers

□ ☑ Assessment

Quiz 1 | Redis streams consumer groups

□ Lesson

Consumer groups

□ ☑ Assessment

Quiz 2 | Redis streams consumer groups

□ Lesson

Adding consumers to a group

□ ☑ Assessment

Quiz 3| Redis streams consumer groups