redis 从入门到实战（2）-redis-cli 命令的使用

作者：SmiteLi

redis-cli

2. redis-cli的两种模式

它有两种主要模式：交互模式，其中有REPL(ReadEvalPrint循环)，用户输入命令并获取回复；另一模式，命令作为redis-cli的参数发送，执行，并打印在标准输出上。

3. redis-cli标准输入输出模式实例

3.1 标准输出

```
$ redis-cli incr mycounter
(integer) 1
$ redis-cli incr mycounter > /tmp/output.txt
$ cat /tmp/output.txt
2
$ redis-cli --raw incr mycounter
3
$ redis-cli --no-raw incr mycounter > /tmp/output.txt
$ cat /tmp/output.txt
(integer) 4
```

3.2 连接指定ip，端口的redis服务器。

默认，redis-cli连接 127.0.0.1:6379。需要连接自定义的redis服务器，如下：

```
$ redis-cli -h redis15.localnet.org -p 6390 -a myUnguessablePazzzzzword123 ping
PONG
```

-h，指定服务器的IP或者域名

-p，指定端口

-a，指定访问密码。

-n，指定访问哪个数据库。默认，redis有16个数据库，从0-15表示。示例如下：

```
$ redis-cli flushall
OK
$ redis-cli -n 1 incr a
(integer) 1
$ redis-cli -n 1 incr a
(integer) 2
$ redis-cli -n 2 incr a
(integer) 1
```

上面的各个参数等同于下面的用 -u <uri>模式：

```
$ redis-cli -u redis://p%40ssw0rd@redis-16379.hosted.com:16379/0 ping
PONG
```

原文链接：redis 从入门到实战（2）-redis-cli 命令的使用

3.3 导入数据

2种方式，一是从标准输入导入，如下：

```
$ redis-cli -x set foo < /etc/services
OK
$ redis-cli getrange foo 0 50
"#\n# Network services, Internet style\n#\n# Note that "
$ redis-cli get foo
```

或者把redis命令放在一个txt文本里：

```
$ cat /tmp/commands.txt
set foo 100
incr foo
append foo xxx
get foo
$ cat /tmp/commands.txt | redis-cli
OK
(integer) 101
(integer) 6
"101xxx"
```

3.4 重复执行同一个命令多次

```
$ redis-cli -r 5 -i 1 incr foo1
(integer) 1
(integer) 2
(integer) 3
(integer) 4
(integer) 5
```

-r，执行的次数

-i，执行命令的间隔

-l，表示持续输出，如下例子：

```
$ redis-cli -r -1 -i 1 INFO | grep rss_human
used_memory_rss_human:1.38M
used_memory_rss_human:1.38M
used_memory_rss_human:1.38M
... a new line will be printed each second ...
```

3.5 输出到csv文件：

```
$ redis-cli lpush mylist a b c d
(integer) 4
$ redis-cli --csv lrange mylist 0 -1
"d","c","b","a"
```

4. redis交互模式示例

```
$ redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> ping helloworld
"helloworld"
127.0.0.1:6379> select 2
OK
127.0.0.1:6379[2]> dbsize
(integer) 0
127.0.0.1:6379[2]> select 0
OK
127.0.0.1:6379> DBSIZE
(integer) 3
127.0.0.1:6379> connect
(error) ERR unknown command `connect`, with args beginning with:
127.0.0.1:6379> connect 127.0.0.1 6379
127.0.0.1:6379> connect 127.0.0.1 9999
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> ping
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> DEBUG restart
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> debug restart
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> ping
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> ping
Could not connect to Redis at 127.0.0.1:9999: Connection refused
not connected> connect 127.0.0.1 6379
127.0.0.1:6379> ping "reconnect"
"reconnect"
127.0.0.1:6379> multi
OK
127.0.0.1:6379> ping
QUEUED
127.0.0.1:6379> exec
1) PONG
127.0.0.1:6379>
```

运行同一命令多次，获取帮助命令：

```
$ redis-cli
127.0.0.1:6379> 5 incr mycounter
(integer) 5
(integer) 6
(integer) 7
(integer) 8
(integer) 9
127.0.0.1:6379> help set

  SET key value [expiration EX seconds|PX milliseconds] [NX|XX]
  summary: Set the string value of a key
  since: 1.0.0
  group: string
```

```
127.0.0.1:6379> help @list

  BLPOP key [key ...] timeout
  summary: Remove and get the first element in a list, or block until one is available
  since: 2.0.0

  BRPOP key [key ...] timeout
  summary: Remove and get the last element in a list, or block until one is available
  since: 2.0.0

  BRPOPLPUSH source destination timeout
  summary: Pop a value from a list, push it to another list and return it; or block until one is av
ilable
  since: 2.2.0

  LINDEX key index
  summary: Get an element from a list by its index
  since: 1.0.0

  LINSERT key BEFORE|AFTER pivot value
  summary: Insert an element before or after another element in a list
  since: 2.2.0

  LLEN key
  summary: Get the length of a list
  since: 1.0.0

  LPOP key
  summary: Remove and get the first element in a list
  since: 1.0.0

  LPUSH key value [value ...]
  summary: Prepend one or multiple values to a list
  since: 1.0.0

  LPUSHX key value
  summary: Prepend a value to a list, only if the list exists
  since: 2.2.0

  LRANGE key start stop
  summary: Get a range of elements from a list
  since: 1.0.0

  LREM key count value
  summary: Remove elements from a list
  since: 1.0.0

  LSET key index value
  summary: Set the value of an element in a list by its index
  since: 1.0.0

  LTRIM key start stop
  summary: Trim a list to the specified range
  since: 1.0.0

  RPOP key
  summary: Remove and get the last element in a list
  since: 1.0.0

  RPOPLPUSH source destination
  summary: Remove the last element in a list, prepend it to another list and return it
  since: 1.2.0

  RPUSH key value [value ...]
  summary: Append one or multiple values to a list
  since: 1.0.0

  RPUSHX key value
  summary: Append a value to a list, only if the list exists
  since: 2.2.0

127.0.0.1:6379>
```

- help @<category> shows all the commands about a given category. The categories are: @eneric, @list, @set, @sorted_set, @hash, @pubsub, @transactions, @connection, @server, @cripting, @hyperloglog.
- help <commandname> shows specific help for the command given as argument.

清楚屏幕信息，输入clear命令。

5. 一些其他操作

5.1 查看redis状态

```
$ redis-cli -i 3 --stat
```

-i，信息输出的时间间隔

5.2 获取一组key

```
$ redis-cli --scan | head -10
key-419
key-71
key-236
key-50
key-38
key-458
key-453
key-499
key-446
key-371
```

匹配指定模式的key：

```
$ redis-cli --scan --pattern '*-11*'
key-114
key-117
key-118
key-113
key-115
key-112
key-119
key-11
key-111
key-110
key-116
```

统计数量：

```
$ redis-cli --scan --pattern 'user:*' | wc -l
3829433
```

5.3 测试延迟：

```
$ redis-cli --latency
min: 0, max: 1, avg: 0.19 (427 samples)
```

```
$ redis-cli --latency-history
min: 0, max: 1, avg: 0.14 (1314 samples) -- 15.01 seconds range
min: 0, max: 1, avg: 0.18 (1299 samples) -- 15.00 seconds range
min: 0, max: 1, avg: 0.20 (113 samples)^C
```

5.4 RDB文件的远程备份

```
$ redis-cli --rdb /tmp/dump.rdb
SYNC sent to master, writing 13256 bytes to '/tmp/dump.rdb'
Transfer finished with success.
```