

Redis Cheatsheet - Basic Commands You Must Know --> UPDATED VERSION --> <https://github.com/LeCoupa/awesome-cheatsheets>

redis_cheatsheet.bash

Raw

```
1 # Redis Cheatsheet
2 # All the commands you need to know
3
4
5 redis-server /path/redis.conf # start redis with the related configuration file
6 redis-cli # opens a redis prompt
7
8
9 # Strings.
10
11
12 APPEND key value # append a value to a key
13 BITCOUNT key [start end] # count set bits in a string
14 SET key value # set value in key
15 SETNX key value # set if not exist value in key
16 SETRANGE key offset value # overwrite part of a string at key starting at the specified offset
17 STRLEN key # get the length of the value stored in a key
18 MSET key value [key value ...] # set multiple keys to multiple values
19 MSETNX key value [key value ...] # set multiple keys to multiple values, only if none of the keys exist
20 GET key # get value in key
21 GETRANGE key value # get a substring value of a key and return its old value
22 MGET key [key ...] # get the values of all the given keys
23 INCR key # increment value in key
24 INCRBY key increment # increment the integer value of a key by the given amount
25 INCRBYFLOAT key increment # increment the float value of a key by the given amount
26 DECR key # decrement the integer value of key by one
27 DECRBY key decrement # decrement the integer value of a key by the given number
28 DEL key # delete key
29
30 EXPIRE key 120 # key will be deleted in 120 seconds
31 TTL key # returns the number of seconds until a key is deleted
32
33
34 # Lists.
35 # A list is a series of ordered values.
36
37
38 RPUSH key value [value ...] # put the new value at the end of the list
39 RPUSHX key value # append a value to a list, only if the exists
40 LPUSH key value [value ...] # put the new value at the start of the list
41 LRANGE key start stop # give a subset of the list
42 LINDEX key index # get an element from a list by its index
43 LINSERT key BEFORE|AFTER pivot value # insert an element before or after another element in a list
44 LLEN key # return the current length of the list
45 LPOP key # remove the first element from the list and returns it
46 LSET key index value # set the value of an element in a list by its index
47 LTRIM key start stop # trim a list to the specified range
48 RPOP key # remove the last element from the list and returns it
49 RPOPLPUSH source destination # remove the last element in a list, prepend it to another list and return it
50 BLPOP key [key ...] timeout # remove and get the first element in a list, or block until one is available
51 BRPOP key [key ...] timeout # remove and get the last element in a list, or block until one is available
52
53
54 # Sets.
55 # A set is similar to a list, except it does not have a specific order and each element may only appear once.
56
57
58 SADD key member [member ...] # add the given value to the set
59 SCARD key # get the number of members in a set
60 SREM key member [member ...] # remove the given value from the set
61 SISMEMBER myset value # test if the given value is in the set.
62 SMEMBERS myset # return a list of all the members of this set
63 SUNION key [key ...] # combine two or more sets and returns the list of all elements
64 SINTER key [key ...] # intersect multiple sets
65 SMOVE source destination member # move a member from one set to another
66 SPOP key [count] # remove and return one or multiple random members from a set
67
68
69 # Sorted Sets
70 # A sorted set is similar to a regular set, but now each value has an associated score.
71 # This score is used to sort the elements in the set.
72
73
74 ZADD key [NX|XX] [CH] [INCR] score member [score member ...] # add one or more members to a sorted set, or update its score if it already
75
76 ZCARD key # get the number of members in a sorted set
77 ZCOUNT key min max # count the members in a sorted set with scores within the given values
78 ZINCRBY key increment member # increment the score of a member in a sorted set
79 ZRANGE key start stop [WITHSCORES] # returns a subset of the sorted set
80 ZRANK key member # determine the index of a member in a sorted set
81 ZREM key member [member ...] # remove one or more members from a sorted set
82 ZREMRANGEBYRANK key start stop # remove all members in a sorted set within the given indexes
83 ZREMRANGEBYSCORE key min max # remove all members in a sorted set, by index, with scores ordered from high to low
84 ZSCORE key member # get the score associated with the given member in a sorted set
85
86 ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count] # return a range of members in a sorted set, by score
87
```

```

88
89 # Hashes
90 # Hashes are maps between string fields and string values, so they are the perfect data type to represent objects.
91
92
93 HGET key field      # get the value of a hash field
94 HGETALL key         # get all the fields and values in a hash
95 HSET key field value # set the string value of a hash field
96 HSETNX key field value # set the string value of a hash field, only if the field does not exist
97
98 HMSET key field value [field value ...] # set multiple fields at once
99
100 HINCRBY key field increment # increment value in hash by X
101 HDEL key field [field ...]  # delete one or more hash fields
102 HEXISTS key field           # determine if a hash field exists
103 HKEYS key                   # get all the fields in a hash
104 HLEN key                    # get all the fields in a hash
105 HSTRLEN key field           # get the length of the value of a hash field
106 HVALS key                   # get all the values in a hash
107
108
109 # HyperLogLog
110 # HyperLogLog uses randomization in order to provide an approximation of the number of unique elements in a set using just a constant, and
111
112
113 PFADD key element [element ...] # add the specified elements to the specified HyperLogLog
114 PFCOUNT key [key ...]           # return the approximated cardinality of the set(s) observed by the HyperLogLog at key's)
115
116 PFMERGE destkey sourcekey [sourcekey ...] # merge N HyperLogLogs into a single one
117
118
119 # Publication & Subscription
120
121
122 PSUBSCRIBE pattern [pattern ...] # listen for messages published to channels matching the given patterns
123 PUBSUB subcommand [argument [argument ...]] # inspect the state of the Pub/Sub subsystem
124 PUBLISH channel message          # post a message to a channel
125 PUNSUBSCRIBE [pattern [pattern ...]] # stop listening for messages posted to channels matching the given patterns
126 SUBSCRIBE channel [channel ...]    # listen for messages published to the given channels
127 UNSUBSCRIBE [channel [channel ...]] # stop listening for messages posted to the given channels
128
129
130 # Other Commands
131
132
133 KEYS pattern # find all keys matching the given pattern

```



Write

Preview

AA B i “ <> ↺ ⋮ ≡ ✓ @ 📌

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

