Redis 常用命令详解

作者：howie404
原文链接：https://ld246.com/article/1573181225236
来源网站：链滴

一、Redis安全设置

编辑Redis的配置文件redis-conf

我的配置文件在:/opt/redis/redis-conf

```
387 # Warning: since Redis is pretty fastan outside user can try up to
388 # 150k passwords per second against agood box. This means that you should
389 # use a very strong password otherwiseit will be very easy to break.
390 #
391 # requirepass foobared
392 requirepass  12345              #在配置文件加上这一行，feige是密码
```

1.1 重启服务

进入/opt/redis/src下，进行：redis-cli shutdown

redis-server/opt/redis/redis-conf & #重启服务，指定配置文件

1.2 redis-cli进入客户端

```
127.0.0.1:6379> KEYS *
(error) NOAUTH Authentication required.              #此时就需要密码验证了
127.0.0.1:6379> auth 12345
OK
127.0.0.1:6379> KEYS *
1) "k5"
2) "k7"
3) "k6"
4) "k4"
5) "k3"
6) "k2"
127.0.0.1:6379> quit
[root@redis redis]# redis-cli -a 12345   #或者使用此种方法验证密码也可以
127.0.0.1:6379> KEYS *
1) "k5"
2) "k7"
3) "k6"
4) "k4"
5) "k3"
6) "k2"
```

二、Redis常用基础命令

2.1 keys命令

```
?    匹配一个字符
*    匹配任意个（包括0个）字符
[]    匹配括号间的任一个字符，可以使用 "-" 符号表示一个范围，如 a[b-d] 可以匹配 "ab","ac","ad"
\x    匹配字符x，用于转义符号，如果要匹配 "?" 就需要使用 \?
```

举例：

```
127.0.0.1:6379> keys *  #查看本库所有的键，默认是库0
1) "settwo"
2) "listone"
3) "listtwo"
4) "usertwo"
5) "userone"
6) "zsetone"
7) "var"
8) "setone"
9) "setthree"
127.0.0.1:6379> keys list*
1) "listone"
2) "listtwo"
```

2.2 判断一个键值是否存在

exists key
如果存在，返回整数类型 1 ，否则返回 0

举例：

```
127.0.0.1:6379> exists userone
(integer) 1
127.0.0.1:6379> exists jj
(integer) 0
```

2.3 删除键

del key [key.....]
可以删除一个或多个键，返回值是删除的键的个数
注意：不支持通配符删除

举例：

```
127.0.0.1:6379> keys *
1) "settwo"
2) "listone"
3) "listtwo"
4) "usertwo"
5) "userone"
6) "zsetone"
7) "var"
8) "setone"
9) "setthree"
127.0.0.1:6379> del var
(integer) 1
127.0.0.1:6379> keys *
1) "settwo"
2) "listone"
3) "listtwo"
4) "usertwo"
5) "userone"
6) "zsetone"
```

7) "setone"
8) "setthree"

2.4 获得键值的数据类型

type key
返回值可能是 string(字符串类型) hash(散列类型) list(列表类型) set(集合类型) zset(有序集合类型)

举例:

127.0.0.1:6379> type setone
set
127.0.0.1:6379> type userone
hash

2.5 切换数据库

127.0.0.1:6379> select 1 #切换到库1，redis默认有16个库0~15
OK

三、Redis中String类型的常用命令

3.1 赋值与取值

set key value 赋值
get key 取值

3.2 递增数值

incr key
当存储的字符串是整数形式时，redis提供了一个使用的命令 incr 作用是让当前的键值递增，并返回增后的值
incr num
当要操作的键不存在时会默认键值为 0 ，所以第一次递增后的结果是 1 ，当键值不是整数时 redis会示错误

3.3 增加指定的整数

incrby key increment
incrby 命令与 incr 命令基本一样，只不过前者可以通过 increment 参数指定一次增加的数值如:
incrby num 2
incrby num 3

3.4 减少指定的整数

decr key
decrby key increment
desc 命令与incr 命令用法相同，只不过是让键值递减
decrby 命令与 incrby命令用法相同

3.5 增加指定浮点数

incrbyfloat key increment
incrbyfloat 命令类似 incrby 命令，差别是前者可以递增一个双精度浮点数，如：
incrbyfloat num 2.7
注意：（受reids 版本限制，版本需要大于 2.6 版本)

3.6 向尾部追加值

append key value

作用是向键值的末尾追加 value ，如果键不存在则将改键的值设置为 value，

即相当于 set key value。返回值是追加后字符串的长度

如：append foo " hello word!"

3.7 获取字符串长度

strlen key
返回键值的长度，如果键不存在则返回0

3.8 同时 获得/设置 多个键值\

mget key [key.....]
mset key value [key value ....]

3.9 位操作

getbit key offset
setbit key offset value
bitcount key [strart] [end]
bitop operation destkey key [key .....]

一个字节由8个二进制位组成，redis 提供了4个命令直接对二进制位进行操作.

getbit 命令

- 可以获得一个字符串类型键指定位置的二进制位的值（0 或 1），索引从 0 开始，
- 如果需要获取的二进制位的索引超出了键值的二进制位的实际长度则默认位值是 0

setbit 命令

- 可以设置字符串类型键指定位置的二进制位的值，返回值是该位置的旧值，如果需要设置的位置过了键值的二进制位的长度
- ，setbit 命令会自动将中间的二进制位设置为0，同理设置一个不存在的键的指定二进制位的值会动将其前面的位赋值为 0

bitcount 命令

- 可以获得字符串类型键中值是1的二进制位个数，可以通过参数来限制统计的字节范围，
- 如我们希望统计前两个字节(即 “aa”）命令：bitcount foo 0 1
- 注意：（受reids 版本限制，版本需要大于 2.6 版本)

bittop 命令

- 可以对多个字符串类型键进行位运算，并将结果存储在destkey参数指定的键中。该命令支持的运操作有
- AND、 OR、XOR、 NOT，如我们对bar 和 aar 进行 OR 运算操作：
- set foo1 bar
- set foo2 aar
- bitop OR res foo1 foo2
- get res
- 注：（受reids 版本限制，版本需要大于 2.6 版本)

四、Redis中Hash类型常用命令

4.1 赋值与取值

```
hset key field value
hget key field
hmset key field value [ field value ..... ]
hmget key field [ field ..... ]
hgetall key
```

- hset 命令用来给字段赋值，而hget命令用来获得字段的值
- hset 命令的方便之处在于不区分插入和更新操作，这意味着修改数据时不用事先判断字段是否存来决定要
- 执行的是插入操作还是更新操 作，当执行的是插入操作时， hset 命令返回 1 ，当执行的是更新作时，
- hset 命令返回的是 0，当键本身不存在时， hset 命令还会自动建立它。
- hmset 设置多个键值
- hmget 获得多个键值
- hgetall 获取键中所有字段和字段值却不知道键中有哪些字段时使用，返回的结果是字段和字段值成的列表

4.2 判断字段是否存在

```
hexists key field
存在返回 1 ， 否则返回 0
```

4.3 当字段不存在时赋值

```
hsetnx key field value
```

hsetnx 命令与hset 命令类似，区别在于如果字段已经存在，hsetnx 命令将不执行任何操作

4.4 增加数字

hincrby key field increment
使字段值增加指定的整数

4.5 删除字段

hdel key field [field .....]
删除一个或多个字段，返回值是被删除的字段个数

4.6 只获取字段名或字段值

hkeys key
hvals key
hkeys 获取所有字段的名字
hvals 获得键中所有字段的值

4.7 获得字段数量

hlen key

五、Redis中List类型常用命令

5.1 向列表两端增加元素

lpush key value [value .....]
rpush key value [value .....]
lpush 命令用来向列表左边增加元素，返回表示增加元素后列表的长度
rpush 命令用来向列表右边增加元素，返回表示增加元素后列表的长度

5.2 从列表两端弹出元素

lpop key
rpop key
lpop 命令可以从列表左边弹出一个元素，lpop 命令执行两步操作，1：将列表左边的元素从列表中除，2：返回被移除元素值
rpop 命令可以从列表右边弹出一个元素

5.3 获取列表中元素个数

llen key
当键不存在时，llen 返回 0

5.4 获得列表片段

lrange key start stop

获得列表中的某一片段，返回索引从 start 到 stop 之间的所有元素（包括两端的元素） 索引开始为 0

注意：

● lrange 与很多语言中用来截取数组片段的方法有一点区别是 lrange 返回的值包含最右边的元素
● lrange 命令也支持负索引，表是从右边开始计算序数，如 ’ -1 ’ 表示最右边第一个元素，’ -2 ’ 表示最右边第二个元素，一次类推。

5.5 删除列表中指定的值

lrem key count value

lrem 命令会删除列表中前 count 个值为 value 的元素，返回值是实际删除的元素个数。
根据count 值的不同，lrem 命令执行的方式会略有差异：

- 当 count > 0 时，lrem 命令会从列表左边开始删除前 count 个值为 value 的元素
- 当 count < 0 时，lrem 命令会从列表右边开始删除前count 个值为 value 的元素
- 当 count = 0 时，lrem 命令会删除所有值为value的元素

5.6 获得 / 设置指定索引的元素值

lindex key index
lset key index value

● lindex 命令用来返回指定索引的元素，索引从 0 开始，
如果 index 是负数则表示从右边开始计算的索引，最右边元素的索引是 -1。
● lset 是通过索引操作列表的命令，它会将索引为 index 的元素赋值为 value

5.7 只保留列表指定片段

ltrim key start end

● ltrim 命令可以删除指定索引范围之外的所有元素，其指定列表范围的方法和 lrange 命令相同.
● ltrim 命令常和 lpush 命令一起使用来限制列表中元素的数量，比如记录日志时我们希望只保留最近的 100 条日志，则每次加入新元素时调用一次ltrim 命令即可.

5.8 向列表中插入元素

linsert key before | after pivot value

linsert 命令首先会在列表中从左到右查找值为 pivot 的元素，然后根据第二个参数是 before 还是 after 来决定将 value 插入到该元素的 前面还是后面,如果命令执行成功，返回插入操作完成之后列表的长度。如果没有找到 pivot 返回 -1 如果key 不存在或为空，返回 0.

5.9 将元素从一个列表转到另一个列表R

rpoplpush source destination

rpoplpush 先执行 rpop 命令在执行 lpush 命令。rpoplpush 命令先会从source 列表类型键的右边弹出一个元素，然后将其加入到 destination 列表类型键的左边，并返回这个元素的值，整过程是原子的.

六、Redis中Set(集合类型)常用命令

6.1 增加删除命令

sadd key member [member ....]
srem key member [member ....]

sadd 命令用来向集合中增加一个或多个元素，如果键不存在则会自动创建。

因为在一个集合中不能有相同的元素，所以如果要加入的元素已经存在与集合中就会忽略这个元素。

返回值是成功加入的元素数量（忽略的元素不计算在内）。

srem 命令用来从集合中删除一个或多个元素，并返回删除成功的个数。

6.2 获得集合中的所有元素

smembers key
返回集合中的所有元素12

6.3 判断元素是否在集合中

sismember key member

判断一个元素是否在集合中是一个时间复杂度为 0(1) 的操作，无论集合中有多少个元素，

sismember 命令始终可以极快的返回结果。当值存在时 sismember 命令返回 1 ，当值不存在或者不存在时返回 0.

6.4 集合间运算

sdiff key [key .....]

sdiff 命令用来对多个集合执行差集运算。集合 A 与集合 B 的差集表示为 A- B ，

代表所有属于 A 且不属于 B 的元素构成的集合,即 A - B = { x| x∈A 且 x ∈/B }.

命令使用方法:

sadd seta 1 2 3 4 6 7 8
qsadd setb 2 3 4
sdiff seta setb

该命令支持同时传入多个键, 计算顺序是先计算 seta 和 setb 在计算结果与 setc 的差集:

sadd setc 2 3 4
sdiff seta setb setc

sinter key [key .....]

该命令用来对多个集合执行交集运算。集合 A 与集合 B 的交集表示为 A∩B，
代表所有属于 A 且属于 B 的元素构成的集合即 A∩B = { x| x∈A 且 x ∈B }.
命令使用方法：

```
sinter seta setb
该命令同样支持同时传入多个键
```

sunion key [key .....]

该命令用来对多个集合执行并集运算。集合 A 与集合 B的并集表示为 A∪B ，
代表所有属于A或所有属于B的元素构成的集合即 A∪B = { x| x∈A 或 x ∈B }.
命令使用方法：

```
sunion seta setb
该命令同样支持同时传入多个键
```

6.5 获得集合中元素的个数

scard key

6.6 进行集合运算并将结果存储

sdiffstore destination key [key .....]
sinterstore destination key [key .....]
sunionstore destination key [key .....]

sdiffstore 命令和 sdiff 命令功能一样，唯一的区别就是前者不会直接返回运算的结果，而是将结果在 destination 键中

sinterstore 这个命令类似于 sinter 命令，但它将结果保存到 destination 集合，而不是简单地返回果集。

sunionstore 这个命令类似于 sunion 命令，但它将结果保存到 destination 集合，而不是简单地返结果集。

6.7 随机获得集合中的元素

srandmember key [count]

该命令用来随机从集合中获取一个元素,还可以传递 count 参数来一次随机获得多个元素，

根据 count 的正负不同，具体表现也不同.当count 为正数时，srandmember 会随机获取从集合里得 count 个不重复的元素。

如果 count 的值大于集合中的元素个数，则srandmember 会返回集合中的全部元素

当 count 为负数时，srandmember 会随机从集合中获得 |count| 个的元素，这些元素有可能相同

注：当传递count 参数时，在windows环境下提示命令参数错误.

6.8 从集合中弹出一个元素

spop key

由于集合类型的元素是无序的，所以 spop 命令会从集合中随机选择一个元素弹出，返回值为被移除随机元素，

如果 key 不存在或者 key 为空集时，返回 nil.

七、Redis中set(有序集合类型（sorted set）)常用命令

7.1 增加元素

zadd key score member [score member .....]

zadd 命令用来向有序集合中加入一个元素和该元素的分数，如果该元素已经存在，则会用新的分数换原有的分数。

zadd命令的返回值是新加入到集合中的元素个数（不包含之前已经存在的元素).

7.2 获得元素的分数

zscore key member1

7.3 获得排名在某个范围的元素列表

zrange key start stop [withscores]
zrevrange key start stop [withscores]

zrange 命令会按照元素分数从小到大的顺序返回索引从 start 到 stop 之间的所有元素（包含两端的素）。

zrange 命令和 lrange 命令十分相似，如索引都是从0开始，负数代表从后向前查找（-1 表示最后一元素）。

如果需要同时获得元素的分数的话，可以在 zrange 命令的尾部加上 widthscores 参数.

注：如果两个元素的分数相同，redis会按照字典顺序（即 0<9

7.4 获得指定分数范围内的元素

zrangebyscore key min max [withscores] [limit offset count]

该命令按照元素分数从小到大的顺序返回分数在 min 到 max 之间（包含 min 和max 的元素)

如果希望分数范围不包含端点值，可以在分数前加上 “(” 符号，例如：希望返回80分到100分的的据，

可以包含80分单不包含100分

命令：zrangebyscore scoreboard 80 (100 widthscores

min 和 max 还支持无穷大，同 zadd 命令一样，-inf 和 +inf 分别表示负无穷大和正无穷大。

比如希望得到所有分数高于 80分（不包含80分）的人的名单，但是却不知道最高分是多少，这是就

以使用 +inf

zrangebyscore scoreboard (80 +inf

命令 limit offset count 与 SQL 中的用法基本相同，即在获得的元素列表的基础上

向后偏移 offset 个元素并且只获取前count个元素zrevrangebyscore 不仅是按照元素分数从大往小顺序给出结果，

而且他的 min 和max 的参数的顺序和 zrangebyscore 命令是相反的。

7.5 增加某个元素的分数

zincrby key increment member

zincrby 命令可以增加一个元素的分数，返回值是更改后的分数，例如想给peter 加 4 分

zincrby scoreborder 4 peter1

increment 也可以是负数表示减分

zincrby scoreborder -4 peter

如果指定元素不存在，redis 在执行命令前会先建立它并将他的值赋为0在执行操作

7.6 获得集合中元素的数量

zcard key

7.7 获得指定分数范围内的元素个数

zcount key min max
zcount 命令的 min max 参数的特性与 zrangebyscore 命令中的一样

7.8 删除一个或多个元素

zrem key member [member ....]
zrem 命令的返回值是成功删除的元素数量（不包含本来就不存在的元素）

7.9 按照排名范围删除元素

zremrangebyrank key start stop
按照元素分数从小到大的顺序（即索引 0 表示最小的值）删除在指定排名范围内的所有元素，并返回除元素的数量

7.10 按照分数范围删除元素

zremrangebyscore key min max1

zremrangebyscore 命令删除指定分数范围内的所有元素，参数 min 和 max 的特性和

zrangebyscore 命令中的一样，返回值是删除元素的个数

7.11 获得元素的排名

zrank key member
zrevrank key member

- zrank 命令会按照元素分数从小到大的顺序获得指定的元素排名（从 0 开始，即分数最小的元素排为0）
- zrebrank 命令则正好相反，分数最大的元素排名为0

7.12 计算有序集合的交集

zinterstore 命令用来计算多个有序集合的交集病将结果存储在 destination 键中（同样以有序集合类存储），

返回值为 destination键中元素的个数，destination 键中元素的分数是由 aggregate 参数决定的.

当 aggregate 是 SUM （也就是默认值），destination 键中元素的分数是每个参与计算的集合中该素分数的和.

当 aggregate 是 MIN 时，destination 键中元素的分数是参与计算的集合中该元素分数最小值

当 aggregate 是 MAX 是，destination 键中元素的分数是参与计算的集合中该元素分数最大值

7.13 计算集合间的并集

zunionstore
用法与 zinterstore 命令的用法一样