Redis的数据类型及操作

李焕贞

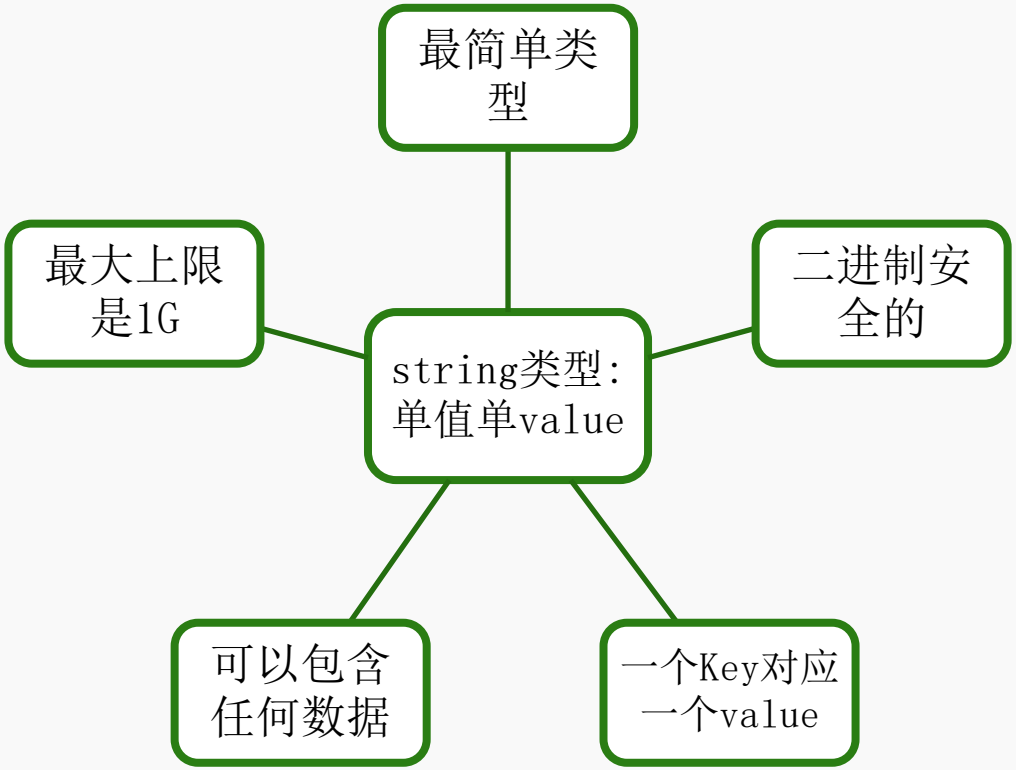
河北师范大学软件学院

http://redisdoc.com/

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中string类型及操作



Redis中string类型及操作

set ： 设置key对应的值为string类型的value

setnx：设置key对应的值为string 类型的value，如果key已经存在返回0，nx是not exist的意思

setex：设置key对应的值为string 类型的value，并指定此键值对应的有效期

```
127.0.0.1:6379> SET name jerry
OK
127.0.0.1:6379> get name
"jerry"
127.0.0.1:6379> SETNX name jack
(integer) 0
127.0.0.1:6379> setex major 10 soft
OK
127.0.0.1:6379> get major
"soft"
```

Redis中string类型及操作

setrange：设置指定的key的value值为子字符串

mset：一次设置多个key的值。成功返回ok表示所有的值都设置了，失败返回0表示没有任何值被设置

```
127.0.0.1:6379> set email lhz@126.com
OK
127.0.0.1:6379> SETRA
(integer) 11
127.0.0.1:6379> get e
"lhz@163.com"
```

```
127.0.0.1:6379> mset email 11@qq.com name tom
OK
127.0.0.1:6379> get email
"11@qq.com"
127.0.0.1:6379> get name
"tom"
```

Redis中string类型及操作

msetnx：一次设置多个key的值。成功返回ok表示所有的值都设置了，失败返回0表示没有任何值被设置，但是不会覆盖已经存在的key

```
127.0.0.1:6379> msetnx email 22@q.com age 20
(integer) 0
```

Redis中string类型及操作

get：获取key对应的string值，如果key不存在返回nil

getset：设置key的值，并返回key的旧值

getrange：获取key的value值的子字符串

mget：一次获取多个key的值，如果对应key不存在则对应返回nil

incr：对key的值做加操作，并返回新的值；key不存在时，会设定key，原来的value是0

incrby：同incr，加指定值，

Redis中string类型及操作

decr：对key值做减操作

decrby：同decr，减指定值

append：给指定key的字符串追加value，返回新字符串值的长度

strlen：取指定key的value值的长度

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中lists类型及操作

list是一个链表结构，主要功能是push、pop、获取一个范围的所有值等等，操作中key理解为链表的名字。list类型其实就是一个每个子元素都是string类型的双向链表。可以通过push、pop操作从链表的头部或者尾部添加删除元素，这样list既可以作为栈，又可以作为队列。

Redis中lists类型及操作

lpush：在key对应的list的头部添加字符串元素

lpop：从list的头部删除元素,并返回删除元素，当key不存在时，返回 nil

```
127.0.0.1:6379> lpush list2 aaa
(integer) 1
127.0.0.1:6379> lpush list2 bbb
(integer) 2
127.0.0.1:6379> LRANGE list2 0 -1
1) "bbb"
2) "aaa"
127.0.0.1:6379> lpop list2
"bbb"
```

Redis中lists类型及操作

rpush：在key对应的list的尾部添加字符串元素

rpop：从list的尾部删除元素,并返回删除元素，当key不存在时，返回 nil

```
127.0.0.1:6379> rpush list1 aa
(integer) 1
127.0.0.1:6379> rpush list1 bb
(integer) 2
127.0.0.1:6379> rpush list1 cc
(integer) 3
127.0.0.1:6379> LRANGE list1 0 -1
1) "aa"
2) "bb"
3) "cc"
```

Redis中lists类型及操作

linsert：在key对应的list的特定位置前或后添加字符串

lset：设置list指定下标的元素值

```
127.0.0.1:6379> LINSERT list1 before bb tom
(integer) 3
127.0.0.1:6379> LRANGE list1 0 -1
1) "aa"
2) "tom"
3) "bb"
127.0.0.1:6379>
```

```
127.0.0.1:6379> lset list2 0 tom
OK
```

Redis中lists类型及操作

lrem：从key对应的list中删除n个和value相同的元素。（n<0从尾部删除，n=0全部删除）

ltrim：保留指定key的值范围内的数据

```
127.0.0.1:6379> LREM list2 0 tom
(integer) 1
```

```
127.0.0.1:6379> rpush list1 aaa bbb ccc
(integer) 3
127.0.0.1:6379> LRANGE list1 0 -1
1) "aaa"
2) "bbb"
3) "ccc"
127.0.0.1:6379> LTRIM list1 1 -1
OK
127.0.0.1:6379> LRANGE list1 0 -1
1) "bbb"
2) "ccc"
```

Redis中lists类型及操作

rpoplpush：从第一个list的尾部移除元素并添加到第二个list的头部

```
127.0.0.1:6379> rpush list1  aaa bbb ccc
127.0.0.1:6379> rpush list2 first second third
(integer) 3
127.0.0.1:6379> RPOPLPUSH list1 list2
"ccc"
127.0.0.1:6379> LRANGE list1 0 -1
1) "aaa"
2) "bbb"
127.0.0.1:6379> LRANGE list2 0 -1
1) "ccc"
2) "first"
3) "second"
4) "third"
```

Redis中lists类型及操作

lindex：返回key的list的index位置的元素

llen：返回key对应的list的长度

```
127.0.0.1:6379> LRANGE list2 0 -1
1) "ccc"
2) "first"
3) "second"
4) "third"
127.0.0.1:6379> LINDEX list2 2
"second"
127.0.0.1:6379> LLEN list2
(integer) 4
```

Redis中lists类型总结

- 它是一个字符串链表，left、right都可以插入添加
- 如果键不存在，创建新的链表
- 如果键已存在，新增内容
- 如果值全移除，对应的键也就消失
- 链表的操作无论是头和尾效率都很高，中间元素操作效率低

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中hash类型及操作

hash是一个string类型的field和value**映射表**。特别适合于**存储对象**，相较于将对象的每个字段存成单个string类型。将一个对象存储在hash类型中会占用更少的内存，并且可以更方便的存取整个对象。

Redis中hash类型及操作

hset:设置hash field为指定值，如果key不存在，则先创建。如果域 field 已经存在于哈希表中， 那么它的旧值将被新值 value 覆盖。

hget: 获取指定的hash field

```
127.0.0.1:6379> hset h1 f1 tom
(integer) 1
127.0.0.1:6379> hset h1 f1 jerry
(integer) 0
127.0.0.1:6379> hget h1 f1
"jerry"
```

Redis中hash类型及操作

hsetnx：设置hash field为指定值，如果key不存在，则先创建。如果存在返回0.

```
127.0.0.1:6379> hsetnx h1 f1 tom
(integer) 1
127.0.0.1:6379> hsetnx h1 f1 tom1
(integer) 0
127.0.0.1:6379> hget h1 f1
"tom"
```

Redis中hash类型及操作

hmset：同时设置hash的多个field

hmget：获取全部指定的hash field

```
127.0.0.1:6379> HMSET h2 name tom age 20

OK
127.0.0.1:6379> HMGET h2 name age
1) "tom"
2) "20"
```

Redis中hash类型及操作

hincrby：指定的hash field加上给定值

hexists：测试指定的field是否存在

hlen：返回指定hash的field数量

```
127.0.0.1:6379> HINCRBY h2 age 5
(integer) 25
127.0.0.1:6379> HGET h2 age
"25"
127.0.0.1:6379> HEXISTS h2 age
(integer) 1
127.0.0.1:6379> HLEN h2
(integer) 2
```

Redis中hash类型及操作

hdel：删除指定的hash的field

hkeys：返回hash的所有field

hvals：返回hash的所有value

```
127.0.0.1:6379> hset user:001 name tom age 20 major soft
(integer) 3
127.0.0.1:6379> HDEL user: 001 age
(integer) 0
127.0.0.1:6379> HKEYS user:001
1) "name"
2) "age"
3) "major"
127.0.0.1:6379> HVALS user:001
1) "tom"
2) "20"
3) "soft"
```

Redis中hash类型及操作

hgetall：获取某个hash中全部的field及value

```
127.0.0.1:6379> HGETALL user:001
1) "name"
2) "tom"
3) "age"
4) "20"
5) "major"
6) "soft"
```

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中set类型及操作

Set是集合，它是string类型的无序集合，不允许有重复的值。set是通过hash table实现的，添加、删除和查找的复杂度都是0(1)。对集合我们可以取并集、交集、差集。通过这些操作我们可以实现sns中的好友推荐和blog的tag功能。

Redis中set类型及操作

sadd：向名称为key的set中添加元素

srem：删除名称为key的set中的某个元素

smembers：返回名称为key的所有元素

```
127.0.0.1:6379> sadd s1 aa
(integer) 1
127.0.0.1:6379> sadd s1 bb
(integer) 1
127.0.0.1:6379> sadd s1 aa
(integer) 0
127.0.0.1:6379> SMEMBERS s1
1) "bb"
2) "aa"
127.0.0.1:6379> SREM s1 bb
(integer) 1
```

Redis中set类型及操作

sdiff：返回所有给定key与第一个key的差集

sdiffstore：返回所有给定key与第一个key的差集，结果存为另一个key

```
127.0.0.1:6379> SMEMBERS s2
1) "a1"
2) "cc"
3) "aa"
127.0.0.1:6379> SMEMBERS s1
1) "cc"
2) "aa"
127.0.0.1:6379> SDIFF s2 s1
1) "a1"
127.0.0.1:6379> SDIFFSTORE s3 s2 s1
(integer) 1
127.0.0.1:6379> SMEMBERS s3
1) "a1"
```

Redis中set类型及操作

sinter：返回所有给定key的交集

sinterstore：返回所有给定key与第一个key的交集，结果存为另一个key

```
127.0.0.1:6379> SADD S1 AA BB CC
(integer) 3
127.0.0.1:6379> SADD S2 BB CC DD
(integer) 3
127.0.0.1:6379> SINTER S1 S2
1) "CC"
2) "BB"
```

```
127.0.0.1:6379> SINTERSTORE s3 s1 s2
(integer) 2
127.0.0.1:6379> SMEMBERS s3
1) "cc"
2) "bb"
```

Redis中set类型及操作

sunion：返回所有给定key并集

sunionstore：返回所有给定key与第一个key的并集，结果存为另一个key

```
127.0.0.1:6379> SUNION s1 s2 s3
1) "ac"
2) "bb"
3) "aa"
4) "cc"
```

Redis中set类型及操作

smove：从第一个key对应的set中移除member并添加到第二个对应的set中

scard：返回名称为key的set的元素个数

sismember：测试member是否是名称为key的set的元素

srandmember：随机返回名称为key的set一个元素，但不删除元素

```
127.0.0.1:6379> SMEMBERS s1
1) "cc"
2) "bb"
3) "aa"
127.0.0.1:6379> SMEMBERS s2
1) "cc"
2) "bb"
3) "ac"
127.0.0.1:6379> SMOVE s2 s1 ac
(integer) 1
```

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中zset类型及操作

sorted set是set的一个升级版本，它在set的基础上增加一个顺序属性，这一属性在添加修改元素的时候可以指定，每次指定后，zset会自动重新按新的值调整顺序。

Redis中hash类型及操作

zadd:向名称为key的zset中添加元素member，score用于排序。如果该元素不存在，则更新其顺序

zrem：删除名称为key的zset中元素

```
127.0.0.1:6379> ZADD z1 1 aa
(integer) 1
127.0.0.1:6379> ZADD z1 1 bb
(integer) 1
127.0.0.1:6379> ZADD z1 2 aa
(integer) 0
127.0.0.1:6379> ZRANGE z1 0 -1 withscores
1) "bb"
2) "1"
3) "aa"
4) "2"
```

Redis中hash类型及操作

zincrby：如果在名称为key的zset中已经存在元素mem，则该元素的score增加increment ，否则向该集合中添加该元素，其socre的值为increment

```
127.0.0.1:6379> ZADD z1 1 aa
(integer) 1
127.0.0.1:6379> ZADD z1 1 bb
(integer) 1
127.0.0.1:6379> ZADD z1 2 aa
(integer) 0
127.0.0.1:6379> ZRANGE z1 0 -1 withscores
1) "bb"
2) "1"
3) "aa"
4) "2"
```

Redis中hash类型及操作

zrank：返回名称为key的member元素的排名（score从小到大的排序），即下标

zrevrank：返回名称为key的member元素的排名（score从大到小的排序），即下标

```
127.0.0.1:6379> ZREVRANK z1 aa
(integer) 1
127.0.0.1:6379> ZRANK z1 aa
(integer) 0
127.0.0.1:6379> zrange z1 0 -1 withscores
1) "aa"
2) "3"
3) "bb"
4) "8"
```

Redis中hash类型及操作

zrange: 返回有序集 key 中，指定区间内的成员

zrevrange:返回有序集 key 中成员 member 的排名。其中有序集成员按 score 值递减(从大到小)排序。

```
127.0.0.1:6379> ZADD z1 1 aa
(integer) 1
127.0.0.1:6379> ZADD z1 1 bb
(integer) 1
127.0.0.1:6379> ZADD z1 2 aa
(integer) 0
127.0.0.1:6379> ZRANGE z1 0 -1 withscores
1) "bb"
2) "1"
3) "aa"
4) "2"
```

Redis中hash类型及操作

zcount：返回分数范围内的成员数量
zcard：返回有序集合中的成员数量

```
127.0.0.1:6379> ZADD z1 1 aa
(integer) 1
127.0.0.1:6379> ZADD z1 1 bb
(integer) 1
127.0.0.1:6379> ZADD z1 2 aa
(integer) 0
127.0.0.1:6379> ZRANGE z1 0 -1 withscores
1) "bb"
2) "1"
3) "aa"
4) "2"
```

本章大纲

- Redis中string类型及操作
- Redis中list类型及操作
- Redis中hash类型及操作
- Redis中set类型及操作
- Redis中zset类型及操作
- Redis中HyperLogLog类型及操作

Redis中HyperLogLog类型及操作

HyperLogLog命令是redis在2.8版本中加入的，Redis中HyperLogLog是用来做基数统计的。

HyperLogLog 的优点是，在输入元素的数量或者体积非常非常大时，计算基数所需的空间总是固定的、并且是很小的，因此每个 HyperLogLog 键只需要花费 12 KB 内存，就可以计算接近 2^{64}个不同元素的基数。

缺点：1、它是估计基数的算法，所以会有一定误差0.81%，而且无法获取具体的元素值。因此应用在对准确性不是很重要的场景，例如：QQ同时在线人数，网站IP访问数量等等。

2、因为 HyperLogLog 只会根据输入元素来计算基数，而不会储存输入元素本身，所以 HyperLogLog 不能像集合那样，返回输入的各个元素。

Redis中HyperLogLog类型及操作

数据集｛1，3，5，7，5，7，8｝，那么这个数据集的基数集为｛1，3，5，7，8｝，基数(不重复元素)为5。 基数估计就是在误差可接受的范围内，快速计算基数。

Redis中HyperLogLog类型及操作

pfadd：将任意数量的元素添加到指定的 HyperLogLog 里面
pfcount：返回给定 HyperLogLog 的基数估算值
pfmerge：将多个 HyperLogLog 合并为一个 HyperLogLog

```
127.0.0.1:6379> PFADD p1  aa bb cc aa
(integer) 1
127.0.0.1:6379> PFCOUNT p1
(integer) 3
127.0.0.1:6379> PFADD p2 tom jerry tom
(integer) 1
127.0.0.1:6379> PFMERGE p3 p1 p2
OK
127.0.0.1:6379> PFCOUNT p3
(integer) 5
```