**Redis五种数据类型介绍**

参考：<http://blog.csdn.net/hechurui/article/details/49508735>

概述

Redis的键值可以使用物种数据类型：**字符串，散列表，列表，集合，有序集合**。本文详细介绍这五种数据类型的使用方法。本文命令介绍部分只是列举了基本的命令，至于具体的使用示例，可以参考Redis官方文档：[Redis命令大全](http://redis.readthedocs.org/en/latest/)

字符串类型

字符串是Redis中最基本的数据类型，它能够存储任何类型的字符串，包含二进制数据。可以用于存储邮箱，JSON化的对象，甚至是一张图片，一个字符串允许存储的最大容量为**512MB**。字符串是其他四种类型的基础，与其他几种类型的区别从本质上来说只是组织字符串的方式不同而已。

基本命令

字符串操作

1. **SET** 赋值，用法： SET key value
2. **GET** 取值，用法： GET key
3. **INCR** 递增数字，仅仅对数字类型的键有用，相当于Java的i++运算，用法： INCR key
4. **INCRBY** 增加指定的数字，仅仅对数字类型的键有用，相当于Java的i+=3，用法：INCRBY key increment，意思是key自增increment，increment可以为负数，表示减少。
5. **DECR** 递减数字，仅仅对数字类型的键有用，相当于Java的i–，用法：DECR key
6. **DECRBY** 减少指定的数字，仅仅对数字类型的键有用，相当于Java的i-=3，用法：DECRBY key decrement，意思是key自减decrement，decrement可以为正数，表示增加。
7. **INCRBYFLOAT** 增加指定浮点数，仅仅对数字类型的键有用，用法：INCRBYFLOAT key increment
8. **APPEND** 向尾部追加值，相当于Java中的”hello”.append(“ world”)，用法：APPEND key value
9. **STRLEN** 获取字符串长度，用法：STRLEN key
10. **MSET** 同时设置多个key的值，用法：MSET key1 value1 [key2 value2 ...]
11. **MGET** 同时获取多个key的值，用法：MGET key1 [key2 ...]

位操作

1. **GETBIT** 获取一个键值的二进制位的指定位置的值(0/1)，用法：GETBIT key offset
2. **SETBIT** 设置一个键值的二进制位的指定位置的值(0/1)，用法：SETBIT key offset value
3. **BITCOUNT** 获取一个键值的一个范围内的二进制表示的1的个数，用法：BITCOUNT key [start end]
4. **BITOP** 该命令可以对多个字符串类型键进行位运算，并将结果存储到指定的键中，BITOP支持的运算包含：**OR,AND,XOR,NOT**，用法：BITOP OP desKey key1 key2
5. **BITPOS** 获取指定键的第一个位值为0或者1的位置，用法：BITPOS key 0/1 [start， end]

散列类型

散列类型相当于Java中的HashMap，他的值是一个字典，保存很多key，value对，每对key，value的值个键都是字符串类型，换句话说，散列类型不能嵌套其他数据类型。一个散列类型键最多可以包含2的32次方-1个字段。

基本命令

1. **HSET** 赋值，用法：HSET key field value
2. **HMSET** 一次赋值多个字段，用法：HMSET key field1 value1 [field2 values]
3. **HGET** 取值，用法：HSET key field
4. **HMGET** 一次取多个字段的值，用法：HMSET key field1 [field2]
5. **HGETALL** 一次取所有字段的值，用法：HGETALL key
6. **HEXISTS** 判断字段是否存在，用法：HEXISTS key field
7. **HSETNX** 当字段不存在时赋值，用法：HSETNX key field value
8. **HINCRBY** 增加数字，仅对数字类型的值有用，用法：HINCRBY key field increment
9. **HDEL** 删除字段，用法：HDEL key field
10. **HKEYS** 获取所有字段名，用法：HKEYS key
11. **HVALS** 获取所有字段值，用法：HVALS key
12. **HLEN** 获取字段数量，用法：HLEN key

列表类型

列表类型(list)用于存储一个有序的字符串列表，常用的操作是向队列两端添加元素或者获得列表的某一片段。列表内部使用的是双向链表（double linked list）实现的，所以向列表两端添加元素的时间复杂度是O(1),获取越接近列表两端的元素的速度越快。但是缺点是使用列表通过索引访问元素的效率太低（需要从端点开始遍历元素）。所以列表的使用场景一般如：朋友圈新鲜事，只关心最新的一些内容。借助列表类型，Redis还可以作为消息队列使用。

基本命令

1. **LPUSH** 向列表左端添加元素，用法：LPUSH key value
2. **RPUSH** 向列表右端添加元素，用法：RPUSH key value
3. **LPOP** 从列表左端弹出元素，用法：LPOP key
4. **RPOP** 从列表右端弹出元素，用法：RPOP key
5. **LLEN** 获取列表中元素个数，用法：LLEN key
6. **LRANGE** 获取列表中某一片段的元素，用法：LRANGE key start stop，index从0开始，-1表示最后一个元素
7. **LREM** 删除列表中指定的值，用法：LREM key count value，删除列表中前count个值为value的元素，当count>0时从左边开始数，count<0时从右边开始数，count=0时会删除所有值为value的元素
8. **LINDEX** 获取指定索引的元素值，用法：LINDEX key index
9. **LSET** 设置指定索引的元素值，用法：LSET key index value
10. **LTRIM** 只保留列表指定片段，用法：LTRIM key start stop，包含start和stop
11. **LINSERT** 像列表中插入元素，用法：LINSERT key BEFORE|AFTER privot value，从左边开始寻找值为privot的第一个元素，然后根据第二个参数是BEFORE还是AFTER决定在该元素的前面还是后面插入value
12. **RPOPLPUSH** 将元素从一个列表转义到另一个列表，用法：RPOPLPUSH source destination

集合类型

集合在概念在高中课本就学过，集合中每个元素都是不同的，集合中的元素个数最多为2的32次方-1个，集合中的元素师没有顺序的。

基本命令

1. **SADD** 添加元素，用法：SADD key value1 [value2 value3 ...]
2. **SREM** 删除元素，用法：SREM key value2 [value2 value3 ...]
3. **SMEMBERS** 获得集合中所有元素，用法：SMEMBERS key
4. **SISMEMBER** 判断元素是否在集合中，用法：SISMEMBER key value
5. **SDIFF** 对集合做差集运算，用法：SDIFF key1 key2 [key3 ...]，先计算key1和key2的差集，然后再用结果与key3做差集
6. **SINTER** 对集合做交集运算，用法：SINTER key1 key2 [key3 ...]
7. **SUNION** 对集合做并集运算，用法：SUNION key1 key2 [key3 ...]
8. **SCARD** 获得集合中元素的个数，用法：SCARD key
9. **SDIFFSTORE** 对集合做差集并将结果存储，用法：SDIFFSTORE destination key1 key2 [key3 ...]
10. **SINTERSTORE** 对集合做交集运算并将结果存储，用法：SINTERSTORE destination key1 key2 [key3 ...]
11. **SUNIONSTORE** 对集合做并集运算并将结果存储，用法：SUNIONSTORE destination key1 key2 [key3 ...]
12. **SRANDMEMBER** 随机获取集合中的元素，用法：SRANDMEMBER key [count]，当count>0时，会随机中集合中获取count个不重复的元素，当count<0时，随机中集合中获取|count|和可能重复的元素。
13. **SPOP** 从集合中随机弹出一个元素，用法：SPOP key

有序集合类型

有序集合类型与集合类型的区别就是他是有序的。有序集合是在集合的基础上为每一个元素关联一个分数，这就让有序集合不仅支持插入，删除，判断元素是否存在等操作外，还支持获取分数最高/最低的前N个元素。有序集合中的每个元素是不同的，但是分数却可以相同。有序集合使用散列表和跳跃表实现，即使读取位于中间部分的数据也很快，时间复杂度为O(log(N))，有序集合比列表更费内存。

基本命令

1. **ZADD** 添加元素，用法：ZADD key score1 value1 [score2 value2 score3 value3 ...]
2. **ZSCORE** 获取元素的分数，用法：ZSCORE key value
3. **ZRANGE** 获取排名在某个范围的元素，用法：ZRANGE key start stop [WITHSCORE]，按照元素从小到大的顺序排序，从0开始编号，包含start和stop对应的元素，WITHSCORE选项表示是否返回元素分数
4. **ZREVRANGE** 获取排名在某个范围的元素，用法：ZREVRANGE key start stop [WITHSCORE]，和上一个命令用法一样，只是这个倒序排序的。
5. **ZRANGEBYSCORE** 获取指定分数范围内的元素，用法：ZRANGEBYSCORE key min max，包含min和max，(min表示不包含min，(max表示不包含max，+inf表示无穷大
6. **ZINCRBY** 增加某个元素的分数，用法：ZINCRBY key increment value
7. **ZCARD** 获取集合中元素的个数，用法：ZCARD key
8. **ZCOUNT** 获取指定分数范围内的元素个数，用法：ZCOUNT key min max，min和max的用法和5中的一样
9. **ZREM** 删除一个或多个元素，用法：ZREM key value1 [value2 ...]
10. **ZREMRANGEBYRANK** 按照排名范围删除元素，用法：ZREMRANGEBYRANK key start stop
11. **ZREMRANGEBYSCORE** 按照分数范围删除元素，用法：ZREMRANGEBYSCORE key min max，min和max的用法和4中的一样
12. **ZRANK** 获取正序排序的元素的排名，用法：ZRANK key value
13. **ZREVRANK** 获取逆序排序的元素的排名，用法：ZREVRANK key value
14. **ZINTERSTORE** 计算有序集合的交集并存储结果，用法：ZINTERSTORE destination numbers key1 key2 [key3 key4 ...] WEIGHTS weight1 weight2 [weight3 weight4 ...] AGGREGATE SUM | MIN | MAX，numbers表示参加运算的集合个数，weight表示权重，aggregate表示结果取值
15. **ZUNIONSTORE** 计算有序几个的并集并存储结果，用法和14一样，不再赘述。

声明

本文为原创，转载请注明出处，本文链接：[http://qifuguang.me/2015/09/29/Redis五种数据类型介绍/](http://qifuguang.me/2015/09/29/Redis%E4%BA%94%E7%A7%8D%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B%E4%BB%8B%E7%BB%8D/)