

# |redis基本使用指令

## keys键操作

- |  |                   |
|--|-------------------|
| 1. <code>exists key</code>             | 判定 key 是否存在       |
| 2. <code>del key1 key2 ... keyN</code> | 删除指定 key          |
| 3. <code>type key</code>               | 返回给定 key 的value类型 |
| 4. <code>keys pattern</code>           | 返回匹配指定模式的所有 key   |
| 5. <code>rename oldname newname</code> | 修改 key 的名字        |
| 6. <code>dbsize</code>                 | 返回当前数据库的 key 的数量  |
| 7. <code>expire key seconds</code>     | 为 key 指定过期时间      |
| 8. <code>ttl key</code>                | 返回 key 的剩余过期秒数    |
| 9. <code>select db_index</code>        | 选择数据库             |
| 10. <code>move key db_index</code>     | 将 key 移动到指定的数据库   |
| 11. <code>flushdb</code>               | 删除当前数据库中所有的 key   |
| 12. <code>flushall</code>              | 删除所有数据库中的所有的key   |

## String类型操作

- |  |                            |
|--|----------------------------|
| 1. <code>set key value</code>                                | 设置 key 对应的值为String类型的value |
| 2. <code>mset key1 value1 key2 value2 ... keyN valueN</code> | 一次设置多个 key 的值              |
| 3. <code>mget key1 key2 ... keyN</code>                      | 一次获得多个 key 的值              |
| 4. <code>incr key</code>                                     | 对key的值做加加操作，并返回新的值         |
| 5. <code>decr key</code>                                     | 对key的值做减减操作，并返回新的值         |
| 6. <code>incrby key Integer</code>                           | 同 incr，加指定值                |
| 7. <code>decrby key Integer</code>                           | 同decr，减指定值                 |
| 8. <code>append key value</code>                             | 给指定 key 的字符串追加 value       |
| 9. <code>substr key start end</code>                         | 返回截取过的 key 的字符串            |

## List类型操作

1. **lpush key** String  
元素  
在 key 对应的 list 的头部（左侧）添加字符串
2. **rpop key**  
元素  
在 list 的尾部（右侧）删除元素，并返回删除
3. **llen key**  
返回 key 对应 list 的长度，key不存在返回0，  
如果key对应的类型不是 list，返回错误
4. **lrange key start end**  
返回指定区间内的元素，下标从0开始
5. **rpush key** string  
元素  
在 key 对应的 list 的尾部（右侧）添加字符串
6. **lpop key**  
元素  
在 list 的头部（左侧）删除元素，并返回删除
7. **ltrim key start end**  
截取 list，保留指定区域内元素

## Set类型操作

1. **sadd key** member  
添加一个String 元素到key 对应的 set 集合  
中，成功返回1,如果元素已在集合中返回0，key不存在返回错误
2. **srem key** member1 member2 ...  
从key对应 set 中移除一个或多个给定元  
素，成功返回移除元素个数
3. **smove key1 key2** member  
从 key1对应的set中移除member，并在  
key2对应的set中添加该元素
4. **scard key**  
返回 set 的元素个数
5. **sismember key** member  
判断 member 是否是在 key 对应的 set 中
6. **sinter key1 key2 ...**  
返回所有给定 key 的交集
7. **sunion key1 key2 ...**  
返回所有给定 key 的并集
8. **sdiff key1 key2 ...**  
返回所哟给定 key 的差集
9. **smembers key**  
返回key对应的set 的所有元素，结果是无序的

## Sort set排序类型操作

1. **zadd key** score member  
score  
添加元素到集合，元素在集合中存在则更新对应

2. **zrem key member** 删除指定元素, 1表示成功, 如果元素不存在则返回0
3. **zeincrby key incrNum member** 按照 incrNum 幅度增加对应的 member 的score, 并返回score
4. **zrank key number** 返回指定元素在集合中的排名的下标, 集合中按score的从小到大排序
5. **zrevrank key number** 同上, 但是集合中是从大到小排序
6. **zrange key start end** 返回集合中从start到end区间内的元素 (包括 start和end) , 返回时有序结果
7. **zrevrange key start end** 同上, 返回结果是按score逆序
8. **zcard key** 返回集合中元素个数
9. **zscore key element** 返回给定元素对应的score
10. **zremrangebyrank key min max** 删除集合中排名在给定区间的元素

## Hash类型

1. **hset key field value** 设置 hash field 为指定值, 如果key 不存在, 则先创建
2. **hget key field** 获得指定的 hash field
3. **hmget key field1 ... fieldN** 获得全部指定的hash field
4. **hmset key field1 value1... fieldN valueN** 同时设置hash的多个field
5. **hincrby key field integer** 将指定的hash field加上指定值
6. **hexists key field** 测试指定field是否存在
7. **hdel key field** 删除指定的hash field
8. **hlen key** 返回指定的hash的field的数量
9. **hkeys key** 返回hash的所有field
10. **hvals key** 返回hash的所有的value
11. **hgetall key** 返回hash 所有的field和value

## redis的持久化相关指令

1. **bgsave** 异步保存数据到磁盘 (保存快照)

2. `lastsave`      返回上一次成功保存到磁盘的Unix时间戳
3. `shutdown`      同步保存到服务器并且关闭redis服务器
4. `bgrewriteaof`    当日志文件过长时优化AOF文件存储

## 以上指令用法

- `./redis-cli bgrewriteaof`
- `./redis-cli lastsave`
- `./redis-cli bgsave`
- `./redis-cli shutdown`
- `./redis-cli -h localhost -p 6379 bgsave`      手动发起快照