redis-cli 命令总结

Redis 提供了丰富的命令（command）对数据库和各种数据类型进行操作，这些 command 可以在 Linux 终端使用。在编程时，比如使用 Redis 的 Java 语言包，这些命令都有对应的方法。下面将 Redis 提供的命令做一总结。

1、连接操作相关的命令

- quit：关闭连接（connection）
- auth：简单密码认证

2、对 value 操作的命令

- exists(key)：确认一个 key 是否存在
- del(key)：删除一个 key
- type(key)：返回值的类型
- keys(pattern)：返回满足给定 pattern 的所有 key
- randomkey：随机返回 key 空间的一个 key
- rename(oldname, newname)：将 key 由 oldname 重命名为 newname，若 newname 存在则删除 newname 表示的 key
- dbsize：返回当前数据库中 key 的数目
- expire：设定一个 key 的活动时间（s）
- ttl：获得一个 key 的活动时间
- select(index)：按索引查询
- move(key, dbindex)：将当前数据库中的 key 转移到有 dbindex 索引的数据库
- flushdb：删除当前选择数据库中的所有 key
- flushall：删除所有数据库中的所有 key

3、对 String 操作的命令

- set(key, value)：给数据库中名称为 key 的 string 赋予值 value
- get(key)：返回数据库中名称为 key 的 string 的 value
- getset(key, value)：给名称为 key 的 string 赋予上一次的 value
- mget(key1, key2,…, key N)：返回库中多个 string（它们的名称为 key1，key2…）的 value
- setnx(key, value)：如果不存在名称为 key 的 string，则向库中添加 string，名称为 key，值为 value
- setex(key, time, value)：向库中添加 string（名称为 key，值为 value）同时，设定过期时间 time

- mset(key1, value1, key2, value2,…key N, value N)：同时给多个 string 赋值，名称为 key i 的 string 赋值 value i
- msetnx(key1, value1, key2, value2,…key N, value N)：如果所有名称为 key i 的 string 都不存在，则向库中添加 string，名称 key i 赋值为 value i
- incr(key)：名称为 key 的 string 增 1 操作
- incrby(key, integer)：名称为 key 的 string 增加 integer
- decr(key)：名称为 key 的 string 减 1 操作
- decrby(key, integer)：名称为 key 的 string 减少 integer
- append(key, value)：名称为 key 的 string 的值附加 value
- substr(key, start, end)：返回名称为 key 的 string 的 value 的子串

4、对 List 操作的命令

- rpush(key, value)：在名称为 key 的 list 尾添加一个值为 value 的元素
- lpush(key, value)：在名称为 key 的 list 头添加一个值为 value 的 元素
- llen(key)：返回名称为 key 的 list 的长度
- lrange(key, start, end)：返回名称为 key 的 list 中 start 至 end 之间的元素（下标从 0 开始，下同）
- ltrim(key, start, end)：截取名称为 key 的 list，保留 start 至 end 之间的元素
- lindex(key, index)：返回名称为 key 的 list 中 index 位置的元素
- lset(key, index, value)：给名称为 key 的 list 中 index 位置的元素赋值为 value
- lrem(key, count, value)：删除 count 个名称为 key 的 list 中值为 value 的元素。count 为 0，删除所有值为 value 的元素，count>0 从 头至尾删除 count 个值为 value 的元素，count<0 从尾到头删除|count|个值为 value 的元素。 lpop(key)：返回并删除名称为 key 的 list 中的首元素 rpop(key)：返回并删除名称为 key 的 list 中的尾元素 blpop(key1, key2,… key N, timeout)：lpop 命令的 block 版本。即当 timeout 为 0 时，若遇到名称为 key i 的 list 不存在或该 list 为空，则命令结束。如果 timeout>0，则遇到上述情况时，等待 timeout 秒，如果问题没有解决，则对 keyi+1 开始的 list 执行 pop 操作。
- brpop(key1, key2,… key N, timeout)：rpop 的 block 版本。参考上一命令。
- rpoplpush(srckey, dstkey)：返回并删除名称为 srckey 的 list 的尾元素，并将该元素添加到名称为 dstkey 的 list 的头部

5、对 Set 操作的命令

- sadd(key, member)：向名称为 key 的 set 中添加元素 member

- srem(key, member) ：删除名称为 key 的 set 中的元素 member
- spop(key) ：随机返回并删除名称为 key 的 set 中一个元素
- smove(srckey, dstkey, member) ：将 member 元素从名称为 srckey 的集合移到名称为 dstkey 的集合
- scard(key) ：返回名称为 key 的 set 的基数
- sismember(key, member) ：测试 member 是否是名称为 key 的 set 的元素
- sinter(key1, key2,…key N) ：求交集
- sinterstore(dstkey, key1, key2,…key N) ：求交集并将交集保存到 dstkey 的集合
- sunion(key1, key2,…key N) ：求并集
- sunionstore(dstkey, key1, key2,…key N) ：求并集并将并集保存到 dstkey 的集合
- sdiff(key1, key2,…key N) ：求差集
- sdiffstore(dstkey, key1, key2,…key N) ：求差集并将差集保存到 dstkey 的集合
- smembers(key) ：返回名称为 key 的 set 的所有元素
- srandmember(key) ：随机返回名称为 key 的 set 的一个元素

6、对 zset（sorted set）操作的命令

- zadd(key, score, member)：向名称为 key 的 zset 中添加元素 member，score 用于排序。如果该元素已经存在，则根据 score 更新该元素的顺序。
- zrem(key, member) ：删除名称为 key 的 zset 中的元素 member
- zincrby(key, increment, member) ：如果在名称为 key 的 zset 中已经存在元素 member，则该元素的 score 增加 increment；否则向集合中添加该元素，其 score 的值为 increment
- zrank(key, member) ：返回名称为 key 的 zset（元素已按 score 从小到大排序）中 member 元素的 rank（即 index，从 0 开始），若没有 member 元素，返回“nil”
- zrevrank(key, member) ：返回名称为 key 的 zset（元素已按 score 从大到小排序）中 member 元素的 rank（即 index，从 0 开始），若没有 member 元素，返回“nil”
- zrange(key, start, end)：返回名称为 key 的 zset（元素已按 score 从小到大排序）中的 index 从 start 到 end 的所有元素
- zrevrange(key, start, end)：返回名称为 key 的 zset（元素已按 score 从大到小排序）中的 index 从 start 到 end 的所有元素
- zrangebyscore(key, min, max)：返回名称为 key 的 zset 中 score >= min 且 score <= max 的所有元素 zcard(key)：返回名称为 key 的 zset 的基数 zscore(key, element)：返回名称为 key 的 zset 中元素 element 的 score zremrangebyrank(key, min, max)：删除名称为 key 的 zset 中 rank >= min 且 rank <= max 的所有元素 zremrangebyscore(key, min, max) ：删除名称为 key 的 zset 中 score >= min 且 score <= max 的所有元素

- zunionstore / zinterstore(dstkeyN, key1,…,keyN, WEIGHTS w1,…wN, AGGREGATE SUM|MIN|MAX)：对 N 个 zset 求并集和交集，并将最后的集合保存在 dstkeyN 中。对于集合中每一个元素的 score，在进行 AGGREGATE 运算前，都要乘以对于的 WEIGHT 参数。如果没有提供 WEIGHT，默认为 1。默认的 AGGREGATE 是 SUM，即结果集合中元素 的 score 是所有集合对应元素进行 SUM 运算的值，而 MIN 和 MAX 是指，结果集合中元素的 score 是所有集合对应元素中最小值和最大值。

7、对 Hash 操作的命令

- hset(key, field, value)：向名称为 key 的 hash 中添加元素 field<—>value
- hget(key, field)：返回名称为 key 的 hash 中 field 对应的 value
- hmget(key, field1, …,field N)：返回名称为 key 的 hash 中 field i 对应的 value
- hmset(key, field1, value1,…,field N, value N)：向名称为 key 的 hash 中添加元素 field i<—>value i
- hincrby(key, field, integer)：将名称为 key 的 hash 中 field 的 value 增加 integer
- hexists(key, field)：名称为 key 的 hash 中是否存在键为 field 的域
- hdel(key, field)：删除名称为 key 的 hash 中键为 field 的域
- hlen(key)：返回名称为 key 的 hash 中元素个数
- hkeys(key)：返回名称为 key 的 hash 中所有键
- hvals(key)：返回名称为 key 的 hash 中所有键对应的 value
- hgetall(key)：返回名称为 key 的 hash 中所有的键（field）及其对应的 value

8、持久化

- save：将数据同步保存到磁盘
- bgsave：将数据异步保存到磁盘
- lastsave：返回上次成功将数据保存到磁盘的 Unix 时戳
- shundown：将数据同步保存到磁盘，然后关闭服务

9、远程服务控制

- info：提供服务器的信息和统计
- monitor：实时转储收到的请求
- slaveof：改变复制策略设置
- config：在运行时配置 Redis 服务器