



Redis Cheat Sheet

By GeekHour @



使用说明

- 红色字体：命令名
- 紫色字体：必要参数
- [棕色字体]：可选参数
- $O(1)$ ：时间复杂度

基本操作

redis-server	启动 Redis
redis-cli (Redis CLI Client)	连接到 Redis
telnet 127.0.0.1 6379 (telnet)	
quit/exit	退出

字符串 (Strings)

SET key value	$O(1)$
设置键key的值为value。	
SETEX key seconds value	$O(1)$
将键key的值设置为value，并且超时时间为seconds秒。	
PSETEX key milliseconds value	$O(1)$
将键key的值设置为value，并且超时时间为milliseconds毫秒。	
SETNX key value	$O(1)$
只有在键key不存在的情况下，将key的值设置为value。	
GET key	$O(1)$
返回键key的值设置为value。	
GETSET key value	$O(1)$
将键key的值为value，并返回设置前的旧值。若没有旧值则返回nil。	
APPEND key value	$O(1)$
如果键key存在并且值是一个字符串，则把value追加到现有值的末尾。 如果key不存在，则将key的值设置为value。 返回追加value之后该value的长度。	$O(N)$
STRANGE key offset value	$O(1)$
从偏移量offset开始用value字符串覆盖键key存储的原字符串。offset从0开始返回修改后新的value字符串的长度。	$O(N)$
GETRANGE key start end	$O(N)$
返回键key存储的字符串的从start到end之间（包括start和end）的部分。 返回value的指定子串。	
INCR/DECR key	$O(1)$
将键key存储的数字值value加一/减一。 若key不存在则值先初始化为0再加一/减一。 若key存储的值不能被解释为数字，则返回错误。	
INCRBY/DECRBY key increment/decrement	$O(1)$
将键key存储的数字值value加上/减去一个increment/decrement的量。 若key不存在则值先初始化为0再加/减。 若key存储的值不能被解释为数字，则返回错误。	
INCRBYFLOAT key increment	$O(1)$
将键key存储的浮点数值value加上一个increment的增量。 若key不存在则值先初始化为0再执行加上增量的操作。 若key存储的值或increment不能被解释为浮点数值，则返回错误。	
MSET key value [key value ...]	$O(N)$
同时为多个键设置值。	
MGET key [key ...]	$O(N)$
返回给定的一个或多个键的值。	
MSETNX key value [key value ...]	$O(N)$
当所有给定的key都不存在时才同时为多个键设置值。 只要有一个key已经存在，那么所有值都不会被继续设置。	
DEL key [key ...]	$O(1)$
删除给定的一个或多个键的值。	
STRLEN key	
获取键为key的值的字符串长度。	

列表 (Lists)

LPUSH key element [element ...]	$O(1)$
将一个或多个元素添加到列表key的表头（左侧），多个值则从左至右依次插入表头； 如果列表key不存在，则创建一个然后执行LPUSH插入操作； 如果列表key存在但不是列表类型，则返回错误。	
LPUSHX key element [element ...]	$O(1)$
当且仅当key存在并且是一个列表的时候，才执行LPUSH操作。	
RPUSH key element [element ...]	$O(1)$
将一个或多个元素添加到列表key的表尾（右侧），多个值则从左至右依次插入表尾； 如果列表key不存在，则创建一个然后执行RPUSH插入操作； 如果列表key存在但不是列表类型，则返回错误。	
RPUSHX key element [element ...]	$O(1)$
当且仅当key存在并且是一个列表的时候，才执行RPUSH操作。	
LPOP key [count ...]	$O(1)$
将一个或count个元素从列表头方向移除并将其返回； 如果列表key不存在，则返回nil。	
RPOP key [count ...]	$O(1)$
将一个或count个元素从列表尾方向移除并将其返回； 如果列表key不存在，则返回nil。	
RPOPLUSH source destination	$O(1)$
在一个原子时间内，将source列表的尾部元素弹出，并插入到destination列表的头部，返回该元素； 如果列表key不存在，则返回nil。	
LREM key count element	$O(N)$
移除列表key中与element相等的count个元素，返回被移除的元素的数量。	
LLEN key	$O(1)$
返回列表key的长度；如果key不存在则返回0；如果key不是列表类型则返回错误。	
LINDEX key index	$O(N)$
返回列表key中索引为index的元素。	
LINSERT key BEFORE AFTER pivot element	$O(N)$
将元素element插入到列表key中，位于pivot之前(BEFORE)或者之后(AFTER)。	
LSET key index element	$O(N)$
将列表key中索引为index的元素设置为element。	
LRANGE key start stop	$O(S+N)$
返回列表key中，位于start和stop之间的元素（包括start和stop）。	
LTRIM key start stop	$O(N)$
只保留列表key中索引为start和stop之间的元素。	
BLPOP key [key ...] timeout	$O(1)$
列表阻塞式(Block)弹出。是LPOP的阻塞版本，当列表中没有任何元素时阻塞，直到超时或发现新的可弹出元素为止。	
BRPOP key [key ...] timeout	$O(1)$
列表阻塞式(Block)弹出。是RPOP的阻塞版本，当列表中没有任何元素时阻塞，直到超时或发现新的可弹出元素为止。	
BRPOPLUSH source destination timeout	$O(1)$
是RPOPLUSH的阻塞版本，当列表source中没有任何元素时阻塞，直到超时或发现新的可弹出元素为止。	

哈希 (Hashes)

HSET key field value [field value ...]	$O(1)$
将哈希表key中的域field的值设置为value。	
HSETNX key field value	$O(1)$
当且仅当field不存在的时候，将哈希表key中的域field的值设置为value。	
HGET key field	$O(1)$
返回哈希表key中给定域field的值。	
HEXISTS key field	$O(1)$
判断给定域field是否存在于哈希表key中。	
HDEL key field [field ...]	$O(1)$
删除哈希表key中的一个或多个指定域field。	
HLEN key	$O(1)$
返回哈希表key中域的数量。	
HSTRLEN key field	$O(1)$
返回哈希表key中，给定域field相关联的值的字符串长度。	
HINCRBY key field increment	$O(1)$
将哈希表key中field的值加上增量increment。	
HINCRBYFLOAT key field increment	$O(1)$
将哈希表key中field的值加上浮点数量增量increment。	
HMSET key field value [field value ...]	$O(N)$
设置多个key-value对。（功能同HSET相同，已过时）。	
HMGET key field [field ...]	$O(N)$
返回哈希表key中一个或多个给定域field的值。	
HKEYS key	$O(N)$
返回哈希表key中的所有域。	
HVALS key	$O(N)$
返回哈希表key中所有域的值。	
HGETALL key	$O(N)$
返回哈希表key中所有的域和值。	
HSCAN key cursor [MATCH pattern] [COUNT count]	
增量迭代，是一个基于游标的迭代器。类似的命令还有SCAN、SSCAN、ZSCAN。	

位图 (Bitmaps)

SETBIT key offset value	$O(1)$
设置或者清除偏移量为offset的位。	
GETBIT key offset	$O(1)$
读取偏移量为offset的位的值。	
BITOP operation destkey key [key ...]	$O(N)$
对一个或者多个key进行位逻辑运算，并将结果存储在destkey中； operation 可以为AND OR XOR NOT。	
BITPOS key bit [start [end [BYTE BIT]]]	$O(N)$
返回第一个bit位（0或者1）的位置。	
BITCOUNT key [start [end [BYTE BIT]]]	$O(N)$
返回从start到end为止的所有设置位（bit为1的位）的数量。	

位域 (Bitfields)

BITFIELD key [GET encoding offset] [OVERFLOW WRAP SAT FAIL] [<SET encoding offset value INCRBY encoding offset increment> [<GET encoding offset>] [OVERFLOW WRAP SAT FAIL] [<SET encoding offset value INCRBY encoding offset increment> ...]]	$O(1)$
将一个String字符串类型当做一个bit数组，执行任意位域整数运算。	
BITFIELD_RO key [GET encoding offset [GET encoding offset ...]]	$O(1)$
将一个String字符串类型当做一个bit数组，执行任意位域整数运算（只读）。	

集合 (Sets)

SADD key member [member ...]	$O(N)$
将一个或多个元素加入到集合key中。已存在于集合中的元素将被忽略。	
SREM key member [member ...]	$O(N)$
将一个或多个元素从集合key中移除。不存在的member元素将被忽略。	
SMEMBERS key	$O(N)$
返回集合key中的所有成员。	
SISMEMBER key member	$O(1)$
判断MEMBER是否是集合key的成员，是返回1，不是或key不存在返回0。	
SPOP key [count]	$O(1)$
移除并返回集合key中的一个或count个随机元素。	
SRANDMEMBER key [count]	$O(1)$
和SPOP类似，区别在于SRANDMEMBER只返回不移除。	
SMOVE source destination member	$O(1)$
将member元素从source集合移动到destination集合。（原子性操作）	
SCARD key	$O(1)$
返回集合key的基数（集合中元素的数量）。	
SCAN cursor [MATCH pattern] [COUNT count] [TYPE type]	
增量迭代，是一个基于游标的迭代器。类似的命令还有HSCAN、SSCAN、ZSCAN。	
SINTER key [key ...]	$O(N*M)$
返回一个集合的全部成员，多个集合则返回交集。	
SINTERSTORE destination key [key ...]	$O(N*M)$
类似SINTER，区别是SINTERSTORE会保存结果集到destination集合中，而不是单纯返回。若destination存在则将其覆盖，destination可以是key本身。	
SUNION key [key ...]	$O(N*M)$
返回一个集合的全部成员，多个集合则返回并集。	
SUNIONSTORE destination key [key ...]	$O(N*M)$
类似SUNION，区别是SUNIONSTORE会保存结果集到destination集合中，而不是单纯返回。若destination存在则将其覆盖，destination可以是key本身。	
SDIFF key [key ...]	$O(N*M)$
返回一个集合的全部成员，多个集合则返回差集。	
SDIFFSTORE destination key [key ...]	$O(N*M)$
类似SDIFF，区别是SDIFFSTORE会保存结果集到destination集合中，而不是单纯返回。若destination存在则将其覆盖，destination可以是key本身。	

有序集合 (SortedSets)

ZADD key [NX XX] [GT LT] [CH] [INCR] score member [score member ...]	$O(M\log N)$
将一个或多个member元素及其分数score加入到有序集合key中。	
ZREM key member [member ...]	$O(N)$
将一个或多个成员从集合key中移除。不存在的member成员将被忽略。	
ZSCORE key member	$O(1)$
返回有序集合key中的成员member的分数值（score）。	
ZINCRBY key increment member	$O(\log N)$
为有序集合key的成员member的分数值score加上一个增量increment。	
ZCARD key	$O(1)$
返回有序集合key的基数（集合中成员的数量）。	
ZCOUNT key min max	$O(1)$
返回有序集合key中，分数值score在min和max之间（包括等于）的成員的数量。	
ZRANGE key start stop [BYSCORE BYLEX] [REV] [LIMIT offset count] [WITHSCORES]	$O(\log N+M)$
返回有序集合key中指定区间内的成员。（从小到大排列）	
ZREVRANGE key start stop [WITHSCORES]	$O(\log N+M)$
返回有序集合key中指定区间内的成员。（从大到小排列）	
ZRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count]	$O(\log N+M)$
返回有序集合key中所有score值在指定区间内的成员。（从小到大排列）	
ZREVRANGEBYSCORE key min max [WITHSCORES] [LIMIT offset count]	$O(\log N+M)$
返回有序集合key中所有score值在指定区间内的成员。（从大到小排列）	
ZRANK key member	$O(\log N)$
返回有序集合key中成员member的排名。（按照score从小到大排列）	
ZREVRANK key member	$O(\log N)$
返回有序集合key中成员member的排名。（按照score从大到小排列）	
ZREMRANGEBYRANK key start stop	$O(\log N+M)$
移除有序集合key中指定排名区间内的所有member。（按照score从小到大排列）	
ZREMRANGEBYSCORE key min max	$O(\log N+M)$
移除有序集合key中指定分数区间内的所有member。（按照score从小到大排列）	
ZREMRANGEBYLEX key min max	$O(\log N+M)$
移除有序集合key中指定字典序区间内的所有member。（按照字典序从小到大排列）	
ZRANGEBYLEX key min max [LIMIT offset count]	$O(\log N)$
返回有序集合key中所有字典序在指定区间内的成员。（从大到小排列）	
ZLEXCOUNT key min max	$O(\log N)$
返回有序集合key中指定字典序区间内的所有member的数量。	
ZSCAN cursor [MATCH pattern] [COUNT count] [TYPE type]	
增量迭代，是一个基于游标的迭代器。类似的命令还有HSCAN、SSCAN、SCAN。	
ZUNIONSTORE destination numkeys key [key ...]	
返回一个或多个有序集合的并集，并存储到destination中。	
ZINTERSTORE destination numkeys key [key ...]	
返回一个或多个有序集合的交集，并存储到destination中。	

流 (Streams)

XADD key [NOMKSTREAM] [MAXLEN MINID [= -] threshold [LIMIT count]] * id field value [field value ...]	$O(1)$
向流追加一个新的消息，如果key不存在则创建。	
XREAD [COUNT count] [BLOCK milliseconds] STREAMS key [key ...] id [id ...]	
读取并返回流key中比id大的消息，若没有则阻塞至消息可用。	
XLEN key	$O(1)$
返回流key中消息的数量。	
XRANGE key start end [COUNT count]	$O(N)$
返回流key中，id位于start和end之间的消息（包括start和end）。	
XINFO STREAM key [FULL [COUNT count]]	$O(1)$
返回流key的信息。	
XACK key group id [id ...]	$O(1)$
返回流key中被消费者组group成功接收的消息数量。	
XCLAIM key group consumer min-idle-time id [id ...]	$O(1)$
返回流key的信息。	
XDEL key id [id ...]	$O(1)$
从流key中移除消息并返回移除消息的数量。	
XGROUP CREATE key groupname id\$ [MKSTREAM] [ENTRIESREAD entries_read]	$O(1)$
创建一个消费者组。	
XGROUP CREATECONSUMER key groupname consumername	$O(1)$
在消费者组groupname中创建一个消费者consumername。	
XGROUP DELCONSUMER key groupname consumername	$O(1)$
从消费者组groupname中删除一个消费者consumername。	
XGROUP DESTROY key groupname	$O(1)$
删除一个消费者组groupname。	
XGROUP SETID key groupname id\$ [ENTRIESREAD entries_read]	$O(1)$
设置一个消费者组groupname的最后交付id。	
XINFO CONSUMERS key groupname	$O(1)$
返回流消费者组kgroupname中的消费者列表。	
XINFO GROUPS key	$O(1)$
返回流key的消费者组列表。	
XINFO STREAM key [FULL [COUNT count]]	$O(1)$
返回流key的信息。	
XTRIM key MAXLEN MINID [= -] threshold [LIMIT count]	$O(1)$
从流的开头删除消息。	

地理空间 (Geospatial)

GEOADD key [NX XX] [CH] longitude latitude member [longitude latitude member ...]	$O(1)$
添加一个或者多个成员到地理空间中。	
GEODIST key member1 member2 [M KM FT MI]	$O(1)$
返回两个成员的距离。	
GEOHASH key member [member ...]	$O(1)$
以哈希字符串的形式返回成员的地理信息。	
GEOPOS key member [member ...]	$O(1)$
返回一个或多个成员的位置经纬度信息。	
GEORADIUS key longitude latitude radius M KM FT MI	$O(N+\log M)$
查询以位置信息(longitude,latitude)为中心，半径为radius范围内的成员，并返回。	
GEORADIUSBYMEMBER key member radius M KM FT MI	$O(N+\log M)$
查询以成员member为中心，半径为radius范围内的成员，并返回。	
GEOSEARCH key FROMMEMBER member FROMLONLAT longitude latitude BYRADIUS radius M KM FT MI BYBOX width height M KM FT MI [ASC DESC] [COUNT count [ANY]] [WITHCOORD] [WITHDIST]	$O(N+\log M)$
指定范围内搜索member并返回。可以按成员（FROMMEMBER），或按经纬度（FROMLONLAT），范围可以是圆形（BYRADIUS）或矩形（BYBOX）。	
GEOSEARCHSTORE destination source FROMMEMBER member FROMLONLAT longitude latitude BYRADIUS radius M KM FT MI BYBOX width height M KM FT MI [ASC DESC] [COUNT count [ANY]] [STOREDIST]	$O(\log N+M)$
与GEOSEARCH相同，指定范围内搜索，并存储结果至destination。	

HyperLogLog

PFADD key [element [element ...]]	$O(1)$
添加一个或者多个元素到HyperLogLog中。若key不存在则创建一个。	
PFCOUNT key [key ...]	$O(1)$
返回一个key的近似的基数。	
PFMERGE destkey sourcekey [sourcekey ...]	$O(N)$
将一个或多个sourcekey合并至destkey。	
PFDEBUG subcommand key	
用来调试HyperLogLog值的一个内部命令。	