### redis 安装

安装：brew install redis

启动：cd /usr/local/Cellar/redis

redis-server

或用后台服务启动：

brew services start redis

安装成功后可以使用客户端程序client-cli和redis服务交互

### 基础

redis有五种数据类型：string,list,hash,set,zset

#### key

键相关命令：

del key //key存在时删除key

keys foo\* //查询所有以foo开头的key

keys \* //查询所有key

exists key //判断key是否存在

expire key time(秒) //为key设置过期时间，time是秒数

expireat key timestamp //设置过期时间，timestamp是秒级时间戳

ttl key //返回key剩余生存时间（秒）

rename key newkey //改key名

type key //获取key对应value的类型

#### string

一个key对应一个value

set key value //设置键值：若key已存在，会覆盖

get key //获取键值

setnx key value //设置键值，若key已存在，会设置失败

mset key1 value1 key2 value2 //同时设置多个key-value

msetnx key1 vlaue1 key2 value2//同时设置多个，只有key不存在时成功

mget key1 key2 //同时获取多个key的值

setex key timeout value //设置key的值，并设置过期时间（秒）（若key存在会覆盖）

incr key //将value值加1

incrby key increment //将value加指定值

decr key //value值减1

decrby key decrement // value值减指定值

getset key newvalue //设置新值，返回旧值

strlen key //返回value长度

getrange key start end //返回从start到end范围内的字符串

append key sth //追加sth到key的value后面

#### hash

适合存储对象

hget key field //获取key中field字段对应的value，若key或field不存在，返回nil

hmget key field1 field2…//获取多个field的值

hset key field value //为key中的field字段赋值value，不存在创建，存在则覆盖

hmset key field1 value3 field2 value4 //设置多个field的value，会覆盖旧值

hgetall key //获取key中所有field及value

hkeys key //获取key中所有filed

hvals key //获取key中所有vlaue

hsetnx key field value //若field存在则设置失败

hincrby key field num //为field加num，num可是负数

hexists key field //判断key中是否存在field

hdel key field1 field2 //删除一个或多个字段

#### list （有序列表）

//将一个值或多个值插入到列表头部(顺序为value1🡪vlaue2)，若key不存在，则会创建，若key已存在，但并不是列表类型，则报错。返回列表的长度

lpush key value1 value2 …

lpop key //移除列表第一个元素

lpushx key value1 value2… //key的列表存在是插入，不存在时插入失败

lrem key count value //移除count个value

count > 0 从表头开始向表尾搜索，移除与 VALUE 相等的元素，数量为 COUNT 。

count < 0 : 从表尾开始向表头搜索，移除与 VALUE 相等的元素，数量为 COUNT 的绝对值。

count = 0 : 移除表中所有与 VALUE 相等的值。

lrange key start end //返回指定区间内的元素

ltrim key start end //对列表进行修剪，只留下区间内的元素（跟lrange区别）

llen key //返回列表长度

lset key index value //通过index索引来更改列表元素的值

lindex key index //通过index获取列表某个元素

rpush key value1 value2 //从列表后端添加一个或多个值

rpop key //从列表后端移除一个元素

rpushx key vlaue1 value2 //为已存在的列表添加一个或多个值

#### set （无序不重复集合）

sadd key vlaue1 value2 //向集合添加一个或多个值，集合中已存在则忽略，返回插入的数量

smembers key // 获取集合中所有元素

srem key value1 value2 //移除集合中一个或多个元素，返回移除的数量

spop key //随机移除一个元素（因为set是无序的），返回被移除的元素

sdiff key1 key2 key3… //返回第一个集合里面不存在于另外集合里面的元素

sinter key1 key2 … //返回多个集合的交集

sunion key1 key2 … //返回过个集合的并集

scard key //返回集合中元素的个数

sismember key value //判断value是否是集合的成员，是返回1，其他返回0

srandmember key [count] //随机返回集合中指定count个元素，count不传返回一个

#### zset

每个元素都关联一个double类型的分数（score），redis正是通过score来为集合成员进行从小到大排序的。

有序集合成员是唯一的，但score是可以重复的。

zadd key score1 value1 score2 value2… //添加元素到有序集合中

zcard key //返回集合中元素数量

zcount key min max //返回score在min~max之间的元素数量

zrange key start end [WITHSCORES] //先按score从小到大排序，start/end是索引，从0开始，withscores参数会把score跟在每个元素后面打印出来

zrevrange key start end [WITHSCORES] //按score从大到小排序，其他同zrange一样

zrangebyscore key min max [withscores] //返回score在min~max范围内的所有成员

默认是开区间（小于等于或大于等于），可以给参数加‘（’来使用闭区间：

zrangebyscore zset (1 5 //1<score<=5

zrank key member //返回某个元素的索引（有序集合按score值从小到大排序）

zrevrank key member //返回某个元素的索引（有序集合按score值从大到小排序）

zscore key member //返回member的score值

zrem key member1 member2… //删除一个或多个元素

zremrangebyrank key start end //移除索引从start~end的元素（按score从小到大排序）

zremrangebyscore key min max //移除score值在min~max之间的元素