Redis是开源的key-value缓存服务产品;提供持久化RDB,AOF，可作为二级缓存使用，替代业务功能不强的mybatis的二级缓存。

## 下载&安装&启动

下载&安装

Windows环境：

下载Redis包，下载地址：：https://github.com/tporadowski/redis/releases。Redis分32位和64位，根据计算机系统下载。下载下来是一个压缩包，解压压缩包，重命名文件夹为redis。

Ubuntu环境

sudo update

sudo apt-get install redis

CentOS环境

将所有操作步骤写成shell文件然后执行（建议），linux环境需具备gcc。shell文件内容如下：

# 下载

wget <http://download.redis.io/releases/redis-6.0.8.tar.gz>

# 解压

tar xzf redis-6.0.8.tar.gz

# 安装

cd redis-6.0.8

make

# 启动服务

cd src

./redis-server

Linux环境下安装时遇到的坑

erver.c:5102:94: error: ‘XXXXXX’ has no member named ‘unixsocket’ serverLog(LL\_NOTICE,"The server is now ready to accept connections at %s", server.unixsocket); ^ server.c:5103:19: error: ‘XXXXXX’ has no member named ‘supervised\_mode’ if (server.supervised\_mode == SUPERVISED\_SYSTEMD) {

更新安装最新的gcc

yum -y install centos-release-scl

yum -y install devtoolset-9-gcc devtoolset-9-gcc-c++ devtoolset-9-binutils

scl enable devtoolset-9 bash

echo "source /opt/rh/devtoolset-9/enable" >>/etc/profile

Could not connect to Redis at 127.0.0.1:6379: Connection refused

退回src上一层目录找到redis.conf 并修改 daemonize no 为 daemonize yes ，默认启动在后台运行

启动

Windows环境，进入Redis目录内，cmd输入：redis-server.exe redis.windows.conf启动redis服务。另开一个cmd窗口，输入redis-cli.exe -h 127.0.0.1 -p 6379使用redis服务。

## Redis概念

Redis默认有

## Redis命令

启动redis：./etc/init.d/redis-server /etc/redis.conf（按照配置文件启动redis）

停止redis：./etc/init.d/redis-server stop

重启redis：./etc/init.d/redis-server restart

进入redis命令框：redis-cli

退出redis命令框：Ctrl + C

Ubuntu系统中修改/etc/redis下的redis.conf配置文件，重启redis

------------------------------------------------------------------------------------------------------------------------------------------------------

type [key] 查看key的数据类型

查key

key \* 查所有key

key \*e 查e结尾的key

key e\* 查e开头的key

exists [key] 查key是否存在，不存在返回0，反之1

move [key] [DBIndex] 将本库的KV移动至DBIndex序号库（0-15库序号），成功返回1，反之0

expire [key] [second]此命令设置key存活时间，默认添加的KV是-1（永久）

ttl [key] 此命令查看key还剩余多少存活时间，过期后返回-2，且通过get命令无法查到

切换当前库：SELECT [0-15] [0-15]库序号

清除当前库：FLUSHDB

清除所有库：FLUSHALL

## 数据类型

|  |  |
| --- | --- |
| 类型 | 解释 |
| String | 字符串 |
| Hash | 类似Java的Map，结构就像JSON：key1:{name:jack,age:18},key2:{name:tom,age:17} |
| List | 列表，允许重复 |
| Set | 无序不重复的KV集 |
| Zset | 有序不重复的集合， 每一个元素都配置一个序号，redis会自动根据算法对序号排序，以此提高查询效率 |

## 数据类型基本操作

### string

set [key] [value] 添加kv（永久，后面可以自己改）

setex [key] [value] [second]添加定时存活的kv

setnx [key] [value] 若key不存在则添加kv

append [key] [value] 向key值尾部拼接value值

strlen [key] 获取key值长度

incr [key] key值累加1

incrby [key] [value] key值累加value

decrby [key] [value] key值累减value

getrange [key] [startIndex] [endIndex] 截取key值

mset […key] 获取多个key值

msetnx [key] [value] …[key n] [value n] 所给定key都不存在时设置多个 key-value 对。

### list

lpush [key] […value] 添加list，【注意：values是倒序压入list中的】

rpush [key] […value] 添加list或新增元素，【注意：values是顺序压入list中的】

lset [key] [index] [value] 设置key[index]值为value

linsert [key] [before|after] [value1] [value2] 在value1值前/后加入value2

lrange [key] [startIndex] [endIndex] 取list范围内数据，startIndex=0,endIndex=-1全取

lpop [key] 栈顶出栈1位元素（头部先出），出后list即少一个数据

rpop [key] 栈底出栈1位元素（尾部先出），出后list即少一个数据

lindex [key] [index] 获取list[index]值【index从0开始】，不改变list

llen [key] list长度

lrem [key] [count] [value] 删除list中count个value值【用于list中重复值较多情况】

ltrim [key] [startIndex] [endIndex] 截取list[startIndex]至list[endIndex]数据保存，其余舍弃。

rpoplpush [key1] [key2] key1中尾部数据弹出一个加入key2头部

### set

sadd [key] […value] 添加/增添【注意，set中不会有重复值，添加的重复值会被过滤掉】

smembers [key] 获取集合中全部值

scard [key] 获取集合元素个数

srem [key] […value] 删除集合中n个value值

srandmember [key] [value] 从集合中随机抽取value个元素

spop [key] 弹出集合元素【涉及到hash值，可认为是随机弹出】，原集合会受到影响

smove [key1] [key2] [value] 将集合key1中的value移动到集合key2中

sdiff [key1] […key] 返回key1与多个key的差集；仅返回key1中的差集。

sinter [key1] […key] 返回key1与多个key的交集；仅返回key1中的交集。

sunion [key1] […key] 返回key1与多个key的并集；仅返回key1中的并集。

### hash

hset [key] [k] [v] 新增/添加

hget [key] [k] 获取值

hmset [key] [k1] [v1]…[kn] [vn] 向hash新增多个kv

hmget [key] [k1]…[kn] 获取多个值

hgetall [key] 获取全部值

hdel [key] [k] 删除

hlen [key] hash的kv数

hexists [key] [k] k是否存在，存在返回1，反之0

hkeys [key] 获取hash中全部key

hvals [key] 获取hash中全部value

hincrby [key] [k] [value] 累加value【value值仅允许整数】

hincrbyfloat [key] [k] [value] 累加value【value值仅允许double型】

hsetnx [key] [k] [value] 若k不存在则新增k value键值对。

### zset

zadd [key] [k1] [score1]…[kn] [scoren] 添加/新增

zrange [key] [startIndex] [endIndex] 取zset范围内k值，startIndex=0,endIndex=-1全取

zrangebyscore [key] [(min|min] [(max|max]获取key中处于[min,max],(min,max)中的元素,‘(’代表不包含。

zrangebyscore [key] [(min|min] [(max|max] limit [startIndex] [count]获取处于值域内的元素，并从开始处截取count个元素

zrem [key] […score] 删除元素

zcard [key] zset长度

zcount [key] [min] [max] 统计处于值域内的元素个数

zrank [key] [k] 获取下标【注意，不是score】

zscore [key] [k] 获取score值

zrevrank [key] [k] 逆序获取下标值

zrevrange [key] [startIndex] [endIndex] 逆序获取所有k。

## 配置密码

config set requirepass "123456"

使用redis前，输入：auth 密码

## 配置“redis限制”

Maxclients

## 配置

查看配置项：命令CONFIG GET *name*（配置项具体名替换name；查看所有配置项，用\*替代name）

修改配置项：命令CONFIG SET *name value*（替换name和value即可）

Redis提供的配置项

include /path/to/local.conf 引用分支配置，位置按情况写

port redis端口

tcp-backlog 511 连接队列，高并发环境下需要一个高backlog值来避免慢客户端连接

bind 127.0.0.1

timeout 0

Tcp-keepalive