

Golang 练习笔记06

Redis使用

[redis的安装](#)

[redis的启动和关闭](#)

[启动redis-server](#)

[启动redis-client](#)

[关闭redis](#)

[redis的基本操作](#)

[使用脚本运行redis](#)

[Go语言使用redis](#)

[使用演示](#)

[运行结果](#)

[参考资料](#)

2020/07/27

[cyc/Golang 练习笔记/Golang 练习笔记06](#)

Redis使用

Redis是一个开源的、使用C语言编写的、支持网络交互的、可基于内存也可持久化的Key-Value数据库。

redis的安装

- 首先在官网[下载redis](#)
- 然后按照官网的提示安装

```
1 $ tar xzf redis-6.0.6.tar
2 $ cd redis-6.0.6
3 $ make
```

redis的启动和关闭

启动redis-server

```
1 $ redis-server
```

启动redis-client

```
1 $ redis-cli
```

关闭redis

```
1 $ redis-cli shutdown
```

- 会直接关闭redis-server
- 当server关闭后client输入东西会显示Connection refused

redis的基本操作

SET key value	设置指定key的值为value
GET key	获取指定key的值(获取不存在的key会返回nil)
GETSET key value	将给定key的值设为value并返回key的旧值(old value)
DEL key	在key存在时删除key
EXISTS key	检查给定key是否存在
RENAME key newkey	修改key的名称为newkey
TYPE key	返回key所储存的值的类型
INCR/DECR key	将key中储存的数字值增/减1 (如果存储的不是数字会报错)
INCRBY/DECR key increment/decrement	将 key 所储存的值加上/减去给定的增量值(increment)/减量值(decrement) (如果存储的不是数字会报错)
EXPIRE key seconds	为给定key设置以秒计的过期时间(过期后key将被删除)
TTL key	以秒为单位，返回给定 key 的剩余生存时间 如果不会过期的key会返回-1 如果key不存在会返回-2

使用脚本运行redis

redis.sh

```
1 #!/bin/sh
2 redis-server
```

运行

```
1 $ chmod +x redis.sh
2 $ ./redis.sh
```

Go语言使用redis

- 获取

```
1 $ go get -v github.com/gomodule/redigo
```

使用演示

reTest.go

```
1 package main
2 import (
3     "fmt"
4     "os"
5     "time"
6     "github.com/gomodule/redigo/redis"
7 )
8
9 func errCheck(info string, err error){
10     if err != nil {
11         fmt.Printf("%s %v\n", info, err)
12         os.Exit(-1)
13     }
14 }
15
```

```

16 func main() {
17     c, err := redis.Dial("tcp", "127.0.0.1:6379")
18     errCheck("Connect to redis error:",err)
19     defer c.Close()
20
21     is_key_exit, err := redis.Bool(c.Do("EXISTS", "mykey"))
22     errCheck("error:", err)
23     fmt.Printf("exists or not: %v \n", is_key_exit)
24
25     _, err = c.Do("SET", "mykey", "hello world")
26     errCheck("redis set failed:", err)
27
28     username, err := redis.String(c.Do("GET", "mykey"))
29     errCheck("redis get failed:", err)
30     fmt.Printf("Get mykey: %v \n", username)
31
32     // 如果获取不存在的值err不会为nil
33     // uncomment下面的代码会导致程序中断
34     // err内容为redigo: nil returned
35     // username1, err := redis.String(c.Do("GET", "mykey1"))
36     // errCheck("redis get failed:", err)
37     // fmt.Printf("Get mykey: %v \n", username1)
38
39     _, err = c.Do("DEL", "mykey")
40     errCheck("redis delete failed:", err)
41
42     _, err = c.Do("SET", "akey", "hello!")
43     errCheck("redis set failed:", err)
44
45     // 设置过期时间为60s
46     n, _ := c.Do("EXPIRE", "akey", 5)
47     if n == int64(1) {
48         fmt.Println("success")
49     }
50
51     timeRemain, err := c.Do("TTL","akey")
52     errCheck("TTL error:",err)
53     fmt.Printf("%d seconds remain\n",timeRemain)
54
55     time.Sleep(8 * 1e9) // sleep for 8 seconds

```

```
56     timeRemain1, err := c.Do("TTL","akey")
57     errCheck("TTL error:",err)
58     // fmt.Printf("The type of timeRemain1 is: %T\n", timeRemain
    1)
59     if timeRemain1 == int64(-2) {
60         fmt.Println("The key is deleted!")
61     } else {
62         fmt.Printf("%d seconds remain\n",timeRemain1)
63     }
64 }
```

运行结果

```
1 $ go run reTest.go
   [ruby-2.6.3p62]
2 exists or not: false
3 Get mykey: hello world
4 success
5 5 seconds remain
6 The key is deleted!
```

参考资料

redis官网

<https://redis.io>

Redis 命令参考

<http://doc.redisfans.com>

菜鸟教程——Redis教程

<https://www.runoob.com/redis/redis-tutorial.html>

Go实战——golang中使用redis(redigo和go-redis/redis)

<https://blog.csdn.net/wangshubo1989/article/details/75050024>

Golang Redis的简单操作

<https://blog.csdn.net/anran186/article/details/90515806>

go语言使用redis (redigo)

<https://www.jianshu.com/p/62f0b9ce7584>

Redis 启动或停止的一些常用命令方法

https://blog.csdn.net/qq_18808965/article/details/81233246

