

Redis的客户端形式

1. 命令行形式 `redis-cli -h ip -p port` (本质上redis也是一个网络服务)
2. 图形化界面, 不一定能连接成功, 中间有跳板机/堡垒机
3. 基于redis 的API 自行开发的客户端 类似于mysql中C语言API和JDBC

重谈Redis的快

相对于MySQL关系型数据库 快!!!

相对于内存中的变量 慢!!!

在使用redis的时候想好应用场景, Redis更适用于分布式系统存储

Redis实战

1. Redis最最最核心的指令

- get 根据key, 获取对应value
- set 把key和对应的value存进去

2. set

[set key value]

3. get 如果key不存在返回nil (类似于null)

[get key]

4. keys : 查询当前服务器上的key, 通过一些符号描述key, 匹配的被查询出来

【key pattern】

5. exists : 判定key是否存在 返回值是key存在的个数。

exists 查询的**时间复杂度**为 $O(1)$, 因为redis组织key按照hash来组织的

```
1 1.exists hello hallo //一次请求, 一次相应
2
3 2.exists hello
4 exists hallo //两次网络请求
```

像上述两种情况，推荐尽量一次请求访问处理多个数据，redis本质是网络服务，网络服务需要通过在网络中传输，需要满足网络对应的要求，需要对请求和响应的报文进行封装和分用，而且网卡属于IO设备慢。

6. del : 删除指定的key

```
del key [keys]
```

7. expire : 给指定的key设置**过期时间** (单位:s) 过期时间: key超出指定的数值, 自动删除

应用场景 验证码、优惠券、基于redis实现分布式锁, 为了避免出现不能正常解锁, 加锁设置福哦起事件

```
expire key seconds 针对已经存在的key
```

8. ttl : time to live 查看key过期时间 返回值三种 时间、-1 没设置、-2没有key

关于redis的key过期策略如何实现?

如何知道谁过期?? 如何知道什么时候过期??

1. 定期删除: 定期删除, 每次抽取一部分, 验证过期时间, 过期的被删除, 保证**抽取检查时间快**

?????? 因为redis单线程, 扫描时间多, 容易阻塞挂掉

2. 惰性删除: 时间到了, key依然存在, 等到下一次访问的时候发现过期, 删除 (类似于小超市买到过期食品)