Redis快速入门

什么是Redis

远程字典服务器，**Re**mote **di**ctionary **s**erver

一个开源的基于**内存存储**的数据库，常用作**键值存储、缓存和消息队列**等

Redis它通常将全部数据存储在内存中，也可以不时（默认两秒）的将数据写入硬盘实现持久化，这里的持久化仅用于重新启动Redis后将数据加载回内存

Redis基本操作

redis默认有16个数据库，默认选择0号数据库

数据库操作

选择数据库

```
select 数据库编号[0,15]
```

添加数据

```
set 键名 值名
```

查看数据库中的键值对数量

```
dbsize
```

清空当前数据库

```
flushdb
```

清空所有数据库

```
flushall
```

将数据保存至磁盘

```
save
```

将数据异步保存至磁盘

```
bgsave
```

获取最后一次成功保存的unix时间

```
lastsave
```

数据操作

查看符合指定格式的key，*为通配符

```
keys 格式
```

查看是否存在指定的key

```
exists 键名...
```

注意：...表示可以有多个！！！

查看指定key对应的value的类型

```
type 键名
```

删除键值对

```
del 键名...
```

重命名键

```
rename k1 k2
```

注意：如果k2存在，则会用原先k1对应的value覆盖k2的value！！！

不覆盖原值重命名

```
renamenx k1 k2
```

按照key将一个键值对移动到指定数据库

```
move 键名 数据库编号
```

按照键名称拷贝value

```
copy 键名1 键名2
```

获取key对应的value

```
get 键名
```

Redis字符串

添加/修改一个键值对

```
set 键名 值名
```

按照key获取value

```
get 键名
```

添加/修改一个或多个键值对

```
mset 键名1 值1 键名2 值2...
```

按照key获取一个或多个value

```
mget 键1 键2...
```

按照key在原有value的基础上追加数据

```
append 键 追加的值
```

按照key查看value的长度

```
strlen 键
```

按照起始索引和key获取对应value的子串

```
getrange 键 开始索引 结束索引
```

注意：在redis中的字符串的索引下标规则和python一致！！！

注意：在redis中执行某个命令希望只有当键不存在时，才进行相关操作，则只需要在命令后加上nx！！！

注意：在redis中执行某个命令希望只有当键存在时，才进行相关操作，则只需要在命令后加上xx！！！

字符串内容为整型数字

按照key给指定value自增1

```
incr 键
```

按照key给指定value增加指定值

```
incr 键 值
```

按照key给指定value自减1

```
decr 键
```

按照key给指定value减少指定值

```
decr 键 值
```

临时键值对

生存时间time to live，缩写为ttl，是指键值对距离被删除的剩余秒数

如果键值对被重新set，则ttl将会被重置

设定生存时间

```
expire 键 秒数
```

查看生存时间剩余秒数

```
ttl 键
```

毫秒版生存时间

```
pexpire 键 毫秒数
```

查看毫秒生存时间所剩余的毫秒数

```
pttl 键
```

取消生存周期(持久化)

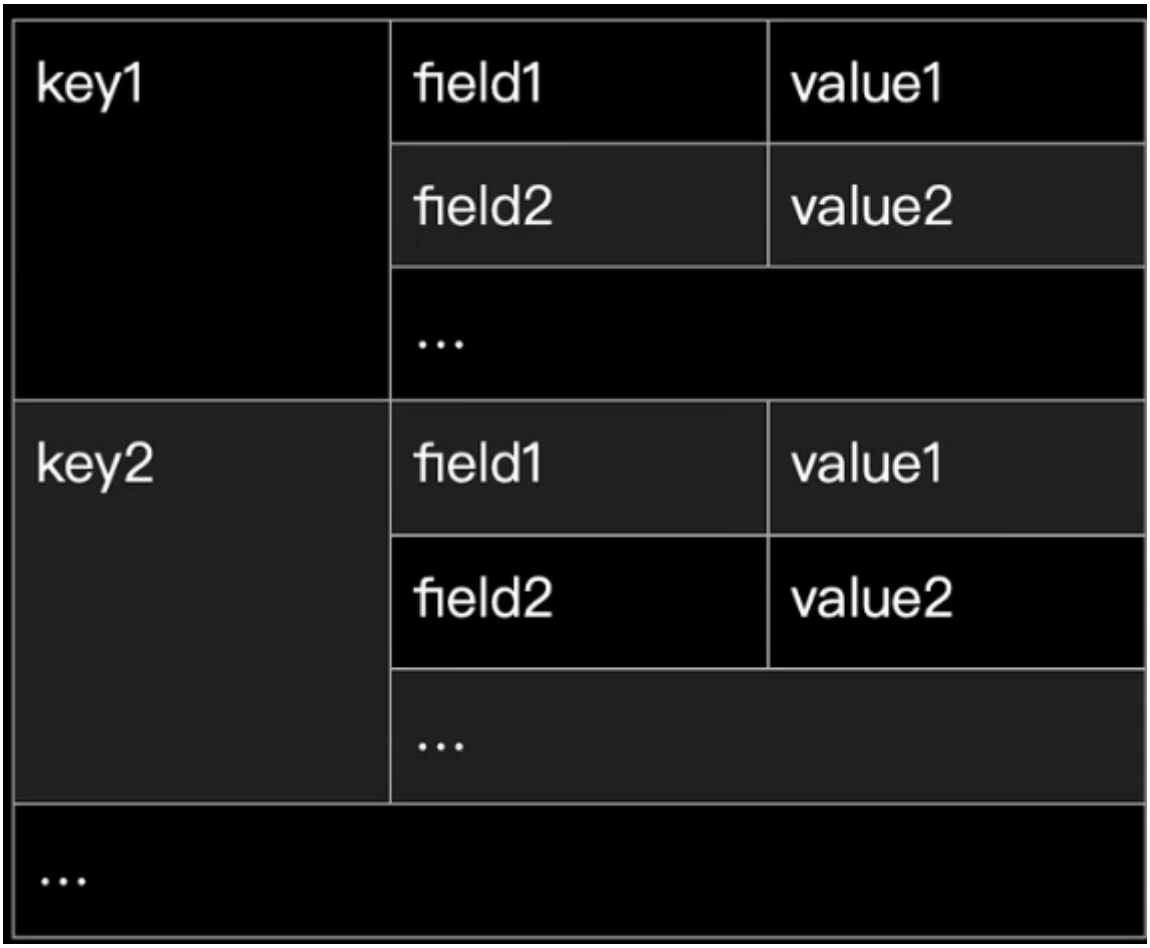
```
persist 键
```

*以下操作仅支持字符串

set key value ex 秒数 setex key 秒数 value	set + expire
set key value px 毫秒数 psetex key 毫秒数 value	set + pexpire
set key value exat unix秒	设置一个unix秒的过期时刻
set key value pxat unix毫秒	设置一个unix毫秒的过期时刻
set key value keepttl	set时不重置ttl

散列表

key-field-value 键-字段-值



Redis散列表基本操作

添加/修改一个键与一至多对字段和值

```
hset 键 field1... value1 ...
```

按照key和field获取一对value

```
hget key field1
```

按照key和field获取多对value

```
hmget key field...
```

按照key获取所有的field和value

```
hgetall key
```

删除一至多对field-value

```
hdel key field1...
```

查看一个散列表中所有的Field

```
hkeys key
```

查看散列表中对应key的所有value

```
hvals key
```

统计一个散列表中某个key下有多少对field-value

```
hlen key
```

查看一个field是否存在

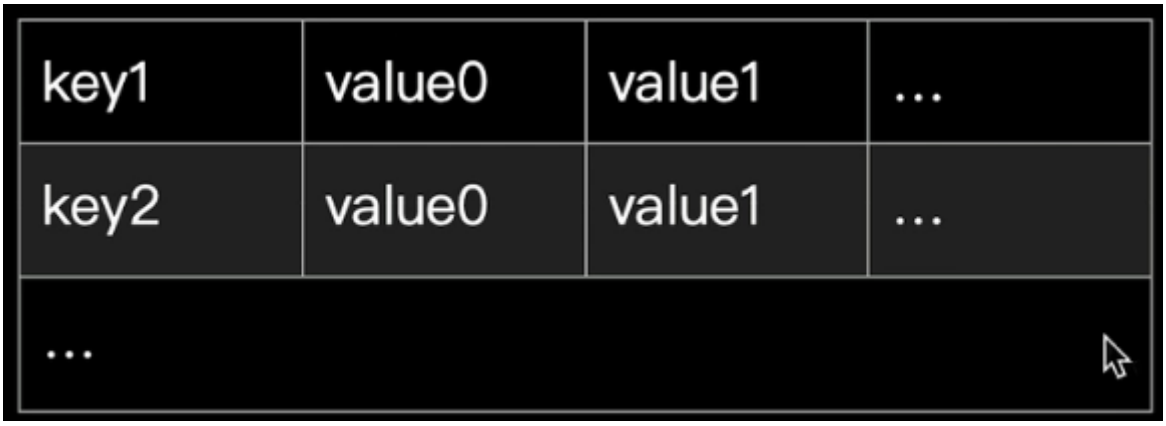
```
hexists key field
```

按照key和filed查看value的长度

```
hstrlen key field
```

Redis列表

数据结构：key-value0-value1-value2-... 键-有序值列队



从右侧插入数据

```
rpush 列表名称 有序value
```

查看列表中的数据

```
lrange 列表名称 开始索引 结束索引
```

从左侧插入数据

```
lpush 列表名称 有序value
```

从右侧弹出数据

```
rpop 列表名称 弹出数据的个数
```

从左侧弹出数据

```
lpop 列表名称 弹出数据的个数
```

修改列表指定位置的值

```
lset 列表名称 索引 新value
```

在列表指定value位置插入指定值

```
linsert 列表名称 after/befor 定位value value
```

按照索引查看值

```
lindex key 索引
```

查看列表长度

```
llen key
```

删除指定个数指定值

```
lrem key 数量 value
```

注意：数量为正数表示从左侧开始删除，数量为负数代表从右侧开始删除

将列表修剪到给定范围

```
ltrim key 开始索引 结束索引
```