**Redis常用命令**

Redis支持五种数据类型：string（字符串），hash（哈希），list（列表），set（集合）及zset(sorted set：有序集合)等

<https://gitee.com/nmwork/RedisUtil>

# Redis数据类型

## List类型

### 简介

Redis列表是简单的字符串列表，按照插入顺序排序。你可以添加一个元素到列表的头部（左边）或者尾部（右边）一个列表最多可以包含 232 - 1 个元素 (4294967295, 每个列表超过40亿个元素)

🡺类似JAVA中的**LinkedList**

### 命令

**赋值语法：**

LPUSH key value1 [value2] //将一个或多个值插入到列表头部(从左侧添加)

RPUSH key value1 [value2] //在列表中添加一个或多个值(从右侧添加)

LPUSHX key value //将一个值插入到已存在的列表头部。如果列表不在，操作无效

RPUSHX key value //一个值插入已存在的列表尾部(最右边)。如果列表不在，操作无效。

**取值语法：**

LLEN key //获取列表长度

LINDEX key index //通过索引获取列表中的元素

LRANGE key start stop  //获取列表指定范围内的元素

描述： 返回列表中指定区间内的元素，区间以偏移量 START 和 END 指定。 其中 0 表示列表的第一个元素， 1 表示列表的第二个元素，以此类推。也可以使用负数下标，以 -1 表示列表的最后一个元素， -2 表示列表的倒数第二个元素，以此类推。

**start: 页大小\*(页数-1)**

**stop : (页大小\*页数)-1**

**删除语法：**

LPOP key 移出并获取列表的第一个元素(从左侧删除)

RPOP key 移除列表的最后一个元素，返回值为移除的元素(从右侧删除)

BLPOP key1 [key2 ] timeout

移出并获取列表的第一个元素， 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。

**实例**

redis 127.0.0.1:6379> BLPOP list1 100

在以上实例中，操作会被阻塞，如果指定的列表 key list1 存在数据则会返回第一个元素，否则在等待100秒后会返回 nil 。

BRPOP key1 [key2 ] timeout

移出并获取列表的最后一个元素， 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。

LTRIM key start stop 对一个列表进行修剪(trim)，就是说，让列表只保留指定区间内的元素，不在指定区间之内的元素都将被删除。

**修改语法：**

LSET key index value 通过索引设置列表元素的值

LINSERT key BEFORE|AFTER world value 在列表的元素前或者后插入元素

描述：将值 value 插入到列表 key 当中，位于值 world 之前或之后。

**高级语法：**

RPOPLPUSH source destination

移除列表的最后一个元素，并将该元素添加到另一个列表并返回   
**示例描述：**

RPOPLPUSH a1 a2 //a1的最后元素移到a2的左侧

RPOPLPUSH a1 a1 //循环列表，将最后元素移到最左侧

BRPOPLPUSH source destination timeout

从列表中弹出一个值，将弹出的元素插入到另外一个列表中并返回它； 如果列表没有元素会阻塞列表直到等待超时或发现可弹出元素为止。

### 应用场景

**项目常应用于：1、对数据量大的集合数据删减 2、任务队列**

1. **对数据量大的集合数据删减**

列表数据显示、关注列表、粉丝列表、留言评价等…分页、热点新闻（Top5)等

利用LRANGE还可以很方便的实现分页的功能,在博客系统中，每片博文的评论也可以存入一个单独的list中。

1. **任务队列**

(list通常用来实现一个消息队列，而且可以确保先后顺序，不必像MySQL那样还需要通过ORDER BY来进行排序)

任务队列介绍(生产者和消费者模式)：

在处理Web客户端发送的命令请求时，某些操作的执行时间可能会比我们预期的更长一些，通过将待执行任务的相关信息放入队列里面，并在之后对队列进行处理，用户可以推迟执行那些需要一段时间才能能完成的操作，这种将工作交给任务处理器来执行的做法被称为任务队列（task queue）。

RPOPLPUSH source destination

移除列表的最后一个元素，并将该元素添加到另一个列表并返回   
**示例描述：**

常用案例：

1:订单系统的下单流程、

2:用户系统登录注册短信等

用户登录成功后，

三个资源：

列表页面 (登录后就允许访问）

实名认证页面(登录后就允许访问）

转帐页面 (登录后、还要实名认证后）

代码演示：



## Set类型

### 简介

Redis 的 Set 是 String 类型的无序集合。集合成员是唯一的，这就意味着集合中不能出现重复的数据。

Redis 中集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。

集合中最大的成员数为 232 - 1 (4294967295, 每个集合可存储40多亿个成员)。

类似于JAVA中的 Hashtable集合

redis的集合对象set的底层存储结构特别神奇，底层使用了intset和hashtable两种数据结构存储的，intset我们可以理解为数组，hashtable就是普通的哈希表（key为set的值，value为null）。

intset内部其实是一个数组（int8\_t coentents[]数组），而且存储数据的时候是有序的，因为在查找数据的时候是通过二分查找来实现的。

### 命令

**赋值语法：**

[SADD key member1 [member2]](http://www.runoob.com/redis/sets-sadd.html) 向集合添加一个或多个成员

**取值语法：**

[SCARD key](http://www.runoob.com/redis/sets-scard.html) 获取集合的成员数

[SMEMBERS key](http://www.runoob.com/redis/sets-smembers.html)  返回集合中的所有成员

[SISMEMBER key member](http://www.runoob.com/redis/sets-sismember.html)  判断 member 元素是否是集合 key 的成员(开发中：验证是否存在判断）

[SRANDMEMBER key [count]](http://www.runoob.com/redis/sets-srandmember.html) 返回集合中一个或多个随机数

**删除语法：**

SREM key member1 [member2] 移除集合中一个或多个成员

SPOP key [count] 移除并返回集合中的一个随机元素

SMOVE source destination member

将 member 元素从 source 集合移动到 destination 集合

**差集语法：**

[SDIFF key1 [key2]](http://www.runoob.com/redis/sets-sdiff.html)  返回给定所有集合的差集(左侧）

SDIFFSTORE destination key1 [key2] 返回给定所有集合的差集并存储在 destination 中

**交集语法：**

[SINTER key1 [key2]](http://www.runoob.com/redis/sets-sinter.html)  返回给定所有集合的交集(共有数据）

SINTERSTORE destination key1 [key2] 返回给定所有集合的交集并存储在 destination 中

**并集语法：**

[SUNION key1 [key2]](http://www.runoob.com/redis/sets-sunion.html) 返回所有给定集合的并集

SUNIONSTORE destination key1 [key2] 所有给定集合的并集存储在 destination 集合中

### 应用场景

**常应用于：对两个集合间的数据[计算]进行交集、并集、差集运算**

1、以非常方便的实现如共同关注、共同喜好、二度好友等功能。对上面的所有集合操作，你还可以使用不同的命令选择将结果返回给客户端还是存储到一个新的集合中。

2、利用唯一性，可以统计访问网站的所有独立 IP

## 有序集合(sorted set) ZSET

### 简介

1、 Redis 有序集合和集合一样也是string类型元素的集合,且不允许重复的成员。

2、不同的是每个元素都会关联一个double类型的分数。redis正是通过分数来为集合中的成员进行从小到大的排序。

3、有序集合的成员是唯一的,但分数(score)却可以重复。

4、集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是O(1)。 集合中最大的成员数为 232 - 1 (4294967295, 每个集合可存储40多亿个成员)。

Redis的ZSet是有序、且不重复

（很多时候，我们都将redis中的有序集合叫做zsets，这是因为在redis中，有序集合相关的操作指令都是以z开头的）

### 命令

**赋值语法：**

ZADD key score1 member1 [score2 member2]

向有序集合添加一个或多个成员，或者更新已存在成员的分数

**取值语法：**

ZCARD key 获取有序集合的成员数

ZCOUNT key min max 计算在有序集合中指定区间分数的成员数

ZRANK key member 返回有序集合中指定成员的索引

ZRANGE key start stop [WITHSCORES]

通过索引区间返回有序集合成指定区间内的成员(低到高)

ZREVRANGE key start stop [WITHSCORES]

返回有序集中指定区间内的成员，通过索引，分数从高到底

**删除语法：**

del key   移除集合

ZREM key member [member ...] 移除有序集合中的一个或多个成员

ZREMRANGEBYRANK key start stop 移除有序集合中给定的排名区间的所有成员(第一名是0)(低到高排序)

ZREMRANGEBYSCORE key min max 移除有序集合中给定的分数区间的所有成员

### 应用场景

**常应用于：排行榜**

1比如twitter 的public timeline可以以发表时间作为score来存储，这样获取时就是自动按时间排好序的。

2比如一个存储全班同学成绩的Sorted Set，其集合value可以是同学的学号，而score就可以是其考试得分，这样在数据插入集合的时候，就已经进行了天然的排序。

3还可以用Sorted Set来做带权重的队列，比如普通消息的score为1，重要消息的score为2，然后工作线程可以选择按score的倒序来获取工作任务。让重要的任务优先执行。