

Redis五大类型:字符串 (String)、哈希/散列/字典 (Hash)、列表 (List)、集合 (Set)、有序集合 (sorted set) 五种

@Resource

RedisTemplate<String, String> redisTemplate;

总括:

redisTemplate.opsForValue();//操作字符串

redisTemplate.opsForHash();//操作hash

redisTemplate.opsForList();//操作list

redisTemplate.opsForSet();//操作set

redisTemplate.opsForZSet();//操作有序set

操作字符串

新增一个字符串类型的值

```
1 redisTemplate.opsForValue().set(key,value)
```

新增一个字符串类型的值和过期时间

```
1 redisTemplate.opsForValue().set(key,value,time)
```

获取key键对应的值

```
1 redisTemplate.opsForValue().get(key)
```

在原有的字符串末尾继续增加字符串

```
1 redisTemplate.opsForValue().append(key, value)
```

截取key值对应的字符串

```
1 redisTemplate.opsForValue().get(key,start,end)
```

获取key值并重新赋值

```
1 redisTemplate.opsForValue().getAndSet(key,value)
```

获取key值得长度

```
1 redisTemplate.opsForValue().size(key)
```

操作Hash

key:关键字

field : 项

value : 值

新增hashMapkey值

```
1 redisTemplate.opsForHash().put(key, filed, value)
```

获取指定key中的值

```
1 redisTemplate.opsForHash().values(key)
```

获取key中的键值对

```
1 redisTemplate.opsForHash().entries(key)
```

获取key中的某项的值

```
1 redisTemplate.opsForHash().get(key, filed)
```

获取key中的所有项

```
1 redisTemplate.opsForHash().keys(key)
```

获取key中长度

```
1 redisTemplate.opsForHash().size(key)
```

以map集合的形式添加

```
1 redisTemplate.opsForHash().putAll(key, new HashMap())
```

新增key的某个项和值，如果key不存在则先新建后添加

```
1 redisTemplate.opsForHash().putIfAbsent(key, filed, value)
```

删除key中某个项，或多个项

```
1 redisTemplate.opsForHash().delete(key, filed, filed.....)
```

操作List

key 键

value 值

index 索引

项：filed

在出现的（第一次出现）项的左边加入值

```
1 redisTemplate.opsForList().leftPush(key, filed, value)
```

向左边批量添加

```
1 redisTemplate.opsForList().leftPushAll(key, value, value....)
```

向左边已存在的集合添加元素

```
1 redisTemplate.opsForList().leftPushIfPresent(key, value)
```

默认使用右边

在出现的（第一次出现）项的右边加入值

```
1 redisTemplate.opsForList().rightPush(key, value, value)
```

向右边批量添加

```
1 redisTemplate.opsForList().rightPushAll(key, value, value...)
```

向右边已存在的集合添加元素

```
1 redisTemplate.opsForList().rightPushIfPresent(key, value)
```

获取指定位置的项的值

```
1 redisTemplate.opsForList().index(key, index)
```

获取指定区间的值

```
1 redisTemplate.opsForList().range(key, start, end)
```

获取长度

```
1 redisTemplate.opsForList().size(key)
```

在指定位置插入值，若存在则覆盖

```
1 redisTemplate.opsForList().set(key, index, value)
```

删除指定位置的项

count 出现的次数，默认为0

```
1 redisTemplate.opsForList().remove(key, count, value)
```

截取指定长度，后保留该长度内的数据

```
1 redisTemplate.opsForList().trim(key, start, end)
```

操作Set

向key中批量添加值

```
1 redisTemplate.opsForSet().add(key, value, value, value...)
```

获取key中的值

```
1 redisTemplate.opsForSet().members(key)
```

获取key中的大小

```
1 redisTemplate.opsForSet().size(key)
```

查看某个值是否存在key中

```
1 redisTemplate.opsForSet().isMember(key, value)
```

批量删除key中的值

```
1 redisTemplate.opsForSet().remove(key, value, value, value...)
```

操作有序Set

score : 分数值 , double类型

max:分数的最大值

min:分数的最小值

向key中添加值和分数值

```
1 redisTemplate.opsForZSet().add(key, value ,score)
```

获取某个区间的值

```
1 redisTemplate.opsForZSet().range(key,start,end)
```

用分数排序

```
1 redisTemplate.opsForZSet().rangeByScore(key,min, max)
```

在分数相同时可以使用某些非分数排序

```
1 redisTemplate.opsForZSet().rangeByLex(key,range)
```

获取个数

```
1 redisTemplate.opsForZSet().count(key,min,max)
```

获取key中某个值得下标

```
1 redisTemplate.opsForZSet().rank(key,value)
```

获取某个值得分数

```
1 redisTemplate.opsForZSet().score(key,value)
```

倒序排列区间的值

```
1 redisTemplate.opsForZSet().reverseRange(key,start,end)
```

获取倒序排列的下标

```
1 redisTemplate.opsForZSet().reverseRank(key,value)
```

批量移除key中的值

```
1 redisTemplate.opsForZSet().remove(key,value,value...)
```

根据分数区间移除值

```
1 redisTemplate.opsForZSet().removeRangeByScore(key,min,max)
```

根据下标移除值

```
1 redisTemplate.opsForZSet().removeRange(key,start,end)
```