

1. 创建 jedis对象

14 `Jedis jedis = new Jedis(String ip , String port)`

2. 键操作

1	清空数据	<code>jedis.flushDB()</code>
2	判断某个键是否存在	<code>boolean jedis.exists(String key)</code>
3	新增键值对 (key,value)	<code>jedis.set(String key , String value)</code>
4	获取所有 key	<code>Set<String> jedis.keys("*")</code>
5	删除键为 key 的数据项	<code>jedis.del(String key)</code>
6	设置键为 key 的过期时间为 i 秒	<code>jedis.expire(String key, int i)</code>
7	获取键为 key 数据项的剩余生存时间 (秒)	<code>int jedis.ttl(String key)</code>
8	移除键为 key 属性项的生存时间限制	<code>jedis.persist(String key)</code>
9	查看键为 key 所对应 value 的数据类型	<code>jedis.type(String key)</code>

3. 字符串操作

1	增加 (或覆盖) 数据项	<code>jedis.set(String key , String value)</code>
2	不覆盖增加数据项 (重复不插入)	<code>jedis.setnx(String key , String value)</code>
3	增加数据项并设置有效时间	<code>jedis.setex(String key , int t , String value)</code>
4	删除键为 key 的数据项	<code>jedis.del(String key)</code>
5	获取键为 key 对应的 value	<code>jedis.get(String key)</code>
6	在 key 对应 value 后面扩展字符串 s	<code>jedis.append(String key , String s)</code>
7	增加多个键值对	<code>jedis.mset(String k1 , String v1 , String k2 , String v2 , ...)</code>
8	获取多个 key 对应 value	<code>String[] jedis.mget(String k1 , String k2 , ...)</code>
9	删除多个 key 对应数据项	<code>jedis.del(new String[]{String k1 , String k2 , ... })</code>
10	获取 key 对应 value 并更新 value	<code>String jedis.getSet(String key , String value)</code>
11	获取 key 对应 value 第 i 到 j 字符	<code>String jedis.getrange(String key , int i , int j)</code>

4. 整数和浮点数操作：

1	增加 (或覆盖) 数据项	<code>jedis.set(String key , String value)</code>
2	获取键为 key 对应的 value	<code>jedis.get(String key)</code>
3	将 key 对应的 value 自加 1	<code>jedis.incr(String key)</code>
4	将 key 对应的 value 自加 n	<code>jedis.incrBy(String key , int n)</code>
5	将 key 对应的 value 自减 1	<code>jedis.decr(String key)</code>
6	将 key 对应的 value 自减 n	<code>jedis.decrBy(String key , int n)</code>

5. 列表 (List) 操作

1	添加一个List	<code>jedis.lpush(String key,String v1,String v2, ...)</code>
2	往key对应List左插入一个元素Vn	<code>jedis.lpush(String key,String Vn)</code>
3	获取key对应List区间[i,j]的元素	<code>jedis.lrange(String key,int i,int j)</code>
4	删除指定元素val个数num	<code>jedis.lrem(String key,int num,String val)</code>
5	删除list区间[i,j]之外的元素	<code>jedis.ltrim(String key,int i,int j)</code>
6	key对应list左出栈一个元素	<code>jedis.lpop(String key)</code>
7	key对应list右插入一个元素Vn	<code>jedis.rpush(String key,String Vn)</code>
8	key对应list右出栈一个元素	<code>jedis.rpop(String key)</code>
9	修改key对应list指定下标index的元素	<code>jedis.lset(String key,int index,String val)</code>
10	获取key对应list的长度	<code>jedis.llen(String key)</code>
11	获取key对应list下标为index的元素	<code>jedis.lindex(String key,int index)</code>
12	把key对应list里面的元素从小到大排序	<code>jedis.sort(String key)</code>

6. 集合 (Set) 操作

1	添加一个Set	<code>jedis.sadd(String key,String v1,String v2, ...)</code>
2	获取key对应set的所有元素	<code>jedis.smembers(String key)</code>
3	删除一个值为val的元素	<code>jedis.srem(String key,String val)</code>
4	删除值为v1,v2,...的元素	<code>jedis.srem(String key,String v...)</code>
5	随机出栈set里的一个元素	<code>jedis.spop(String key)</code>
6	获取set中元素个数	<code>jedis.scard(String key)</code>
7	将元素val从集合k1剪切到k2	<code>jedis.smove(String k1,String k2,String val)</code>
8	获取集合k1和集合k2的交集	<code>jedis.sinter(String k1,String k2)</code>
9	获取集合k1和集合k2的并集	<code>jedis.sunion(String k1,String k2)</code>
10	获取集合k1和集合k2的差集	<code>jedis.sdiff(String k1,String k2)</code>

7. 哈希 (Hash) 操作

1	添加一个Hash	<code>jedis.hmset(String key,Map map)</code>
2	往Hash插入一个元素(K-V)	<code>jedis.hset(String key,String key,String value)</code>
3	获取Hash的所有(K-V)元素	<code>jedis.hgetAll(String key)</code>
4	获取Hash所有元素的key	<code>jedis.hkeys(String key)</code>
5	获取Hash所有元素的value	<code>jedis.hvals(String key)</code>
6	把Hash中k对应元素val+=i	<code>jedis.hincrBy(String key,String k,int i)</code>
7	把Hash中k对应元素val-=i	<code>jedis.hdecrBy(String key,String k,int i)</code>
8	从Hash删除一个或多个元素	<code>jedis.hdel(String key,String k1,String k2,...)</code>
9	获取Hash中元素个数	<code>jedis.hlen(String key)</code>
10	判断Hash是否存在k1对应元素	<code>jedis.hexists(String key,String k1)</code>
11	获取Hash中一个或多个元素value	<code>jedis.hmget(String key,String k1,String k2,...)</code>

8. 有序集合 (Zsort) 操作

1	添加一个ZSet	jedis.zadd(String key, Map map)
2	往ZSet插入一个元素(Score-Val)	jedis.hset(String key, int score, String val)
3	获取ZSet里下标[i,j]区间元素val	jedis.zrange(String key, int i, int j)
4	获取ZSet里下标[i,j]区间元素(Score-Val)	jedis.zrangeWithScores(String key, int i, int j)
5	获取ZSet里score[i,j]区间元素val	jedis.zrangeByScore(String key, int i, int j)
6	获取ZSet里score[i,j]区间元素(Score-Val)	jedis.zrangeByScoreWithScores(String key,int i,int j)
7	获取ZSet里val为value元素的score	jedis.zscore(String key, String value)
8	获取ZSet里val为value元素的score排名	jedis.zrank(String key, String value)
9	删除ZSet里val为value的元素	jedis.zrem(String key, String value)
10	获取ZSet的元素个数	jedis.zcard(String key)
11	获取ZSet中score在[i,j]区间元素个数	jedis.zcount(String key, int i, int j)
12	把ZSet中val为value的元素score+=n	jedis.zincrby(String key, int n, String value)

9. 排序操作

1	生成一个排序对象 (SortingParams)	new SortingParams();
2	队列按首字母a-z排序	jedis.sort(String key, sortingParams.alpha())
3	队列按数字升序排序	jedis.sort(String key, sortingParams.asc())
4	队列按数字降序排序	jedis.sort(String key, sortingParams.desc())

升序排列和降序排列仅可用于value值是数值类型

默认排序是升序排列