http357029540.iteye.comblog2389045

[**RedisTemplate常用集合使用说明-opsForHash(四)**](http://357029540.iteye.com/blog/2389045)

**博客分类：**

* [spring boot](http://357029540.iteye.com/category/364646)
* [RedisTemplate](http://357029540.iteye.com/category/374804)

[springboot](http://www.iteye.com/blogs/tag/springboot)[RedisTemplate](http://www.iteye.com/blogs/tag/RedisTemplate)[Spring](http://www.iteye.com/blogs/tag/Spring)[opsForHash](http://www.iteye.com/blogs/tag/opsForHash)

     基础配置介绍已经在前面的《[RedisTemplate常用集合使用说明(一)](http://357029540.iteye.com/blog/2388706)》中已经介绍了，现在我们直接介绍opsForHash()方法的使用：

**1、**[**put**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#put-H-HK-HV-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**hashKey,**[**HV**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**value)**

   新增hashMap值。

redisTemplate.opsForHash().put("hashValue","map1","map1-1");

redisTemplate.opsForHash().put("hashValue","map2","map2-2");

**2、**[**values**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#values-H-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key)**

  获取指定变量中的hashMap值。

List<Object> hashList = redisTemplate.opsForHash().values("hashValue");

System.out.println("通过values(H key)方法获取变量中的hashMap值:" + hashList);

**3、**[**entries**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#entries-H-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key)**

  获取变量中的键值对。

Map<Object,Object> map = redisTemplate.opsForHash().entries("hashValue");

System.out.println("通过entries(H key)方法获取变量中的键值对:" + map);

**4、**[**get**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#get-H-java.lang.Object-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html?is-external=true)**hashKey)**

  获取变量中的指定map键是否有值,如果存在该map键则获取值，没有则返回null。

Object mapValue = redisTemplate.opsForHash().get("hashValue","map1");

System.out.println("通过get(H key, Object hashKey)方法获取map键的值:" + mapValue);

**5、**[**hasKey**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#hasKey-H-java.lang.Object-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html?is-external=true)**hashKey)**

      判断变量中是否有指定的map键。

boolean hashKeyBoolean = redisTemplate.opsForHash().hasKey("hashValue","map3");

System.out.println("通过hasKey(H key, Object hashKey)方法判断变量中是否存在map键:" + hashKeyBoolean);

**6、**[**keys**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#keys-H-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key)**

       获取变量中的键。

Set<Object> keySet = redisTemplate.opsForHash().keys("hashValue");

System.out.println("通过keys(H key)方法获取变量中的键:" + keySet);

**7、**[**size**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#size-H-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key)**

        获取变量的长度。

long hashLength = redisTemplate.opsForHash().size("hashValue");

System.out.println("通过size(H key)方法获取变量的长度:" + hashLength);

**8、**[**increment**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#increment-H-HK-double-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**hashKey, double delta)**

       使变量中的键以double值的大小进行自增长。

double hashIncDouble = redisTemplate.opsForHash().increment("hashInc","map1",3);

System.out.println("通过increment(H key, HK hashKey, double delta)方法使变量中的键以值的大小进行自增长:" + hashIncDouble);

**9、**[**increment**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#increment-H-HK-long-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**hashKey, long delta)**

使变量中的键以long值的大小进行自增长。

long hashIncLong = redisTemplate.opsForHash().increment("hashInc","map2",6);

System.out.println("通过increment(H key, HK hashKey, long delta)方法使变量中的键以值的大小进行自增长:" + hashIncLong);

**10、**[**multiGet**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#multiGet-H-java.util.Collection-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**Collection**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html?is-external=true)**<**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**> hashKeys)**

         以集合的方式获取变量中的值。

List<Object> list = new ArrayList<Object>();

list.add("map1");

list.add("map2");

List mapValueList = redisTemplate.opsForHash().multiGet("hashValue",list);

System.out.println("通过multiGet(H key, Collection<HK> hashKeys)方法以集合的方式获取变量中的值:"+mapValueList);

**11、**[**putAll**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#putAll-H-java.util.Map-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**Map**](https://docs.oracle.com/javase/6/docs/api/java/util/Map.html?is-external=true)**<? extends**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**,? extends**[**HV**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**> m)**

   以map集合的形式添加键值对。

Map newMap = new HashMap();

newMap.put("map3","map3-3");

newMap.put("map5","map5-5");

redisTemplate.opsForHash().putAll("hashValue",newMap);

map = redisTemplate.opsForHash().entries("hashValue");

System.out.println("通过putAll(H key, Map<? extends HK,? extends HV> m)方法以map集合的形式添加键值对:" + map);

**12、**[**putIfAbsent**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#putIfAbsent-H-HK-HV-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**HK**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**hashKey,**[**HV**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**value)**

          如果变量值存在，在变量中可以添加不存在的的键值对，如果变量不存在，则新增一个变量，同时将键值对添加到该变量。

redisTemplate.opsForHash().putIfAbsent("hashValue","map6","map6-6");

map = redisTemplate.opsForHash().entries("hashValue");

System.out.println("通过putIfAbsent(H key, HK hashKey, HV value)方法添加不存在于变量中的键值对:" + map);

**13、**[**scan**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#scan-H-org.springframework.data.redis.core.ScanOptions-)**(**[**H**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[**ScanOptions**](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/ScanOptions.html)**options)**

            匹配获取键值对，ScanOptions.NONE为获取全部键对，ScanOptions.scanOptions().match("map1").build()     匹配获取键位map1的键值对,不能模糊匹配。

Cursor<Map.Entry<Object,Object>> cursor = redisTemplate.opsForHash().scan("hashValue",ScanOptions.scanOptions().match("map1").build());

//Cursor<Map.Entry<Object,Object>> cursor = redisTemplate.opsForHash().scan("hashValue",ScanOptions.NONE);

while (cursor.hasNext()){

Map.Entry<Object,Object> entry = cursor.next();

System.out.println("通过scan(H key, ScanOptions options)方法获取匹配键值对:" + entry.getKey() + "---->" + entry.getValue());

}

**14、**[delete](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html#delete-H-java.lang.Object...-)**(**[H](http://docs.spring.io/spring-data/redis/docs/current/api/org/springframework/data/redis/core/HashOperations.html)**key,**[Object](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html?is-external=true)**... hashKeys)**

   删除变量中的键值对，可以传入多个参数，删除多个键值对。

redisTemplate.opsForHash().delete("hashValue","map1","map2");

map = redisTemplate.opsForHash().entries("hashValue");

System.out.println("通过delete(H key, Object... hashKeys)方法删除变量中的键值对后剩余的:" + map);