13 实验 SpringBoot操作Redis常用数据类型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 1910705127 | 姓名 | 宋宇昕 | 班级 | 19软件1班 |
| 上课地点 | 精工楼 | 教师 | 孟昭睿 | 上课时间 | 12.13 |

ReaderVO实体类

package com.example.codetwo.domain;  
  
import lombok.Data;  
  
import java.util.Date;  
  
@Data  
public class ReaderVo {  
 public static final String *Table* = "t\_reader";  
  
 private String name;  
 private String address;  
 private Integer age;  
 private double account;  
 private boolean ifUse;  
 private Integer sex;  
 private Date date;  
}

Zset，lisl，Hash部分

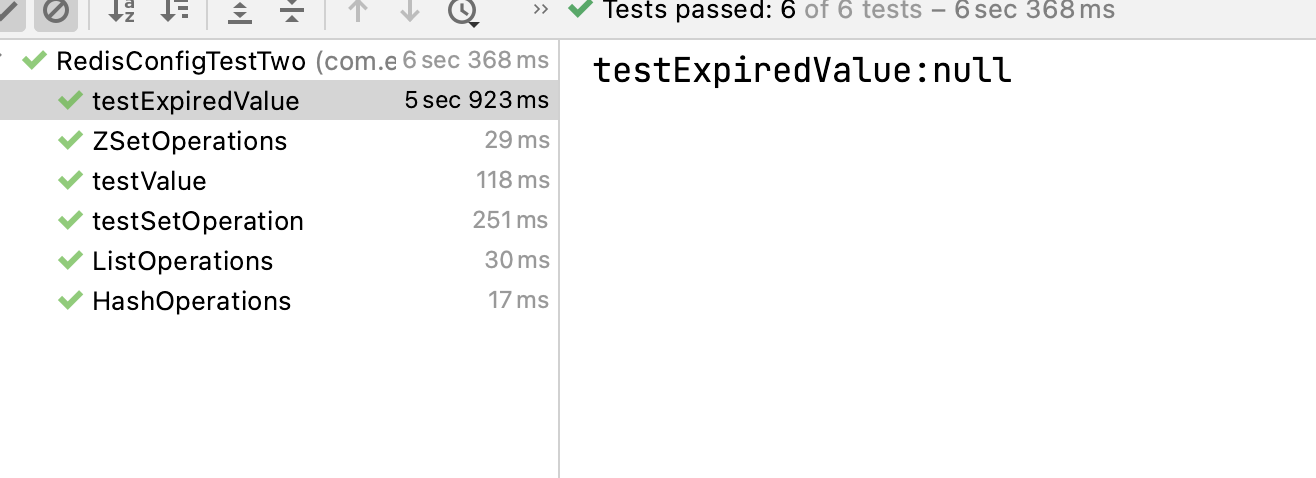
@Test  
public void testExpiredValue() throws Exception {  
 ReaderVo readerVo = new ReaderVo();  
 Date date = new Date();  
 readerVo.setName("花无缺");  
 readerVo.setAccount(145.2);  
 readerVo.setSex(1);  
 readerVo.setAge(18);  
 readerVo.setAddress("山西省");  
 readerVo.setIfUse(true);  
 readerVo.setDate(date);  
 valueOperations.set("reader",readerVo,3, TimeUnit.*SECONDS*);  
 Thread.*currentThread*().*sleep*(5000);  
 System.*out*.println("testExpiredValue:" + valueOperations.get("reader"));  
  
}

@Test  
public void testValue() throws Exception {  
 ReaderVo readerVo = new ReaderVo();  
 Date date = new Date();  
 readerVo.setName("花无缺");  
 readerVo.setAccount(145.2);  
 readerVo.setSex(1);  
 readerVo.setAge(18);  
 readerVo.setAddress("山西省");  
 readerVo.setIfUse(true);  
 readerVo.setDate(date);  
 valueOperations.set("reader",readerVo);  
 System.*out*.println("testValue:" + valueOperations.get("reader"));  
  
}

@Test  
public void testSetOperation() throws Exception {  
 ReaderVo readerVo = new ReaderVo();  
 Date date = new Date();  
 readerVo.setName("花无缺");  
 readerVo.setAccount(145.2);  
 readerVo.setSex(1);  
 readerVo.setAge(18);  
 readerVo.setAddress("山西省");  
 readerVo.setIfUse(true);  
 readerVo.setDate(date);  
 ReaderVo auserVo = new ReaderVo();  
 auserVo.setName("清风");  
 auserVo.setAccount(128.2);  
 auserVo.setSex(0);  
 auserVo.setAge(8);  
 auserVo.setAddress("陕西省");  
 auserVo.setIfUse(true);  
 auserVo.setDate(date);  
 setOperations.add("reader:test", readerVo, readerVo,auserVo);  
 Set<Object> result = setOperations.members("reader:test");  
 System.*out*.println("testSetOperation: " + result);  
}

@Test  
public void HashOperations() throws Exception {  
 ReaderVo readerVo = new ReaderVo();  
 Date date = new Date();  
 readerVo.setName("花无缺");  
 readerVo.setAccount(145.2);  
 readerVo.setSex(1);  
 readerVo.setAge(18);  
 readerVo.setAddress("山西省");  
 readerVo.setIfUse(true);  
 readerVo.setDate(date);  
 hashOperations.put("hash:reader", readerVo.hashCode() + "", readerVo);  
 System.*out*.println("HashOperations: " + hashOperations.get("hash:reader", readerVo.hashCode() + ""));  
}

@Test  
public void ListOperations() throws Exception {  
 ReaderVo readerVo = new ReaderVo();  
 Date date = new Date();  
 readerVo.setName("花无缺");  
 readerVo.setAccount(145.2);  
 readerVo.setSex(1);  
 readerVo.setAge(18);  
 readerVo.setAddress("山西省");  
 readerVo.setIfUse(true);  
 readerVo.setDate(date);  
 listOperations.leftPush("list:reader", readerVo);  
 System.*out*.println("ListOperations: " + listOperations.leftPop("list:reader"));  
 *// pop之后 值会消失* System.*out*.println("ListOperations: " + listOperations.leftPop("list:reader"));  
}



@Test  
public void ZSetOperations() throws Exception {  
 ZSetOperations.TypedTuple<Object> objectTypedTuple1 = new DefaultTypedTuple<Object>("zSet-1",9.6);  
 ZSetOperations.TypedTuple<Object> objectTypedTuple2 = new DefaultTypedTuple<Object>("zSet-6",9.9);  
 Set<ZSetOperations.TypedTuple<Object>> tuples = new HashSet<ZSetOperations.TypedTuple<Object>>();  
 tuples.add(objectTypedTuple1);  
 tuples.add(objectTypedTuple2);  
 System.*out*.println("ZSetOperations: " +zSetOperations.add("zSet-1",tuples));  
 System.*out*.println("ZSetOperations: " +zSetOperations.range("zSet-6",0,-1));  
}