使用spring-data-redis实现incr自增

应该有不少人在使用spring-data-redis时遇到各种各样的问题，下面列举一些：

**redis.clients.jedis.exceptions.JedisDataException: ERR value is not an integer or out of range**

使用的”RedisTemplate”做读写操作的时候，都是要经过序列化和反序列化。这时你使用”redisTemplate.opsForValue().incremnt()”就可能报错”redis.clients.jedis.exceptions.JedisDataException: ERR value is not an integer or out of range”了。

valueOper.get(key)获取不到自增的值。看了一下redis的官方文档，找到了一个解决方法：

使用spring-data-redis实现incr自增

/\*\*

\*

\* @param key

\* @param liveTime

\* @return

\*/

public Long incr(String key, long liveTime) {

RedisAtomicLong entityIdCounter = new RedisAtomicLong(key, redisTemplate.getConnectionFactory());

Long increment = entityIdCounter.getAndIncrement();

if ((null == increment || increment.longValue() == 0) && liveTime > 0) {//初始设置过期时间

entityIdCounter.expire(liveTime, TimeUnit.SECONDS);

}

return increment;

}

这样，上面的increment就是自增后的新值，然后中间通过“entityIdCounter.expire(liveTime, TimeUnit.SECONDS);”设置过期时间。当然这里比较讨厌，spring没有在创建RedisAtomicLong对象的时候一起设置过期时间。可以看看其源码，“new RedisAtomicLong”最终调用的是这个方法：

private RedisAtomicLong(String redisCounter, RedisConnectionFactory factory, Long initialValue) {

Assert.hasText(redisCounter, "a valid counter name is required");

Assert.notNull(factory, "a valid factory is required");

RedisTemplate<String, Long> redisTemplate = new RedisTemplate<String, Long>();

redisTemplate.setKeySerializer(new StringRedisSerializer());

redisTemplate.setValueSerializer(new GenericToStringSerializer<Long>(Long.class));

redisTemplate.setExposeConnection(true);

redisTemplate.setConnectionFactory(factory);

redisTemplate.afterPropertiesSet();

this.key = redisCounter;

this.generalOps = redisTemplate;

this.operations = generalOps.opsForValue();

if (initialValue == null) {

if (this.operations.get(redisCounter) == null) {

set(0);

}

} else {

set(initialValue);

}

}

可以看到，初始值是0.然后跟进set方法：

public void set(long newValue) {

operations.set(key, newValue);

}

可以看到，它是采用的”operations.set(key, newValue);”但是明明还有一个重载的方法”void set(K key, V value, long timeout, TimeUnit unit);”可以设置过期时间，为啥spring不提供呢？为了解决这个问题，我们可以自己模拟”RedisAtomicLong”方法，去实现一个带有过期时间的自增方法。比较简单，大家可以自己发挥。