Redis-03-Jedis 和 SpringBoot 整合

作者：Anileh

原文链接：https://ld246.com/article/1652098679563

来源网站：链滴

1. 准备工作

所需jar包

```
<dependency>
<groupId>redis.clients</groupId>
<artifactId>jedis</artifactId>
<version>3.2.0</version>
</dependency>
```

连接redis的注意事项

禁用Linux的防火墙：systemctl stop / disable firewalld.service

redis.conf中注释掉 bind 127.0.0.1，然后 protected-mode改成no

2 常用操作

2.1 创建测试程序

```
public class JedisTest {
    Jedis jedis = new Jedis("192.168.137.3",   6379);
    jedis.atth("password");
    String pong = jedis.ping():
    System.out.println("连接成功" + pong):
    jedis.close():
}
```

2.2 Jedis-API: Key

```
jedis.set("k1", "v1");
jedis.set("k2", "v2");
jedis.set("k3", "v3");
Set<String> keys = jedis.keys("*");
System.out.println(keys.size());
for (String key : keys) {
System.out.println(key);
}
System.out.println(jedis.exists("k1"));
System.out.println(jedis.ttl("k1"));
System.out.println(jedis.get("k1"));
```

2.3 Jedis-API: String

```
jedis.mset("str1","v1","str2","v2","str3","v3");
System.out.println(jedis.mget("str1","str2","str3"));
```

2.4 Jedis-API: List

```
List<String> list = jedis.lrange("mylist",0,-1);
for (String element : list) {
System.out.println(element);
}
```

2.5 Jedis-API: Set

```
jedis.sadd("orders", "order01");
jedis.sadd("orders", "order02");
jedis.sadd("orders", "order03");
jedis.sadd("orders", "order04");
Set<String> smembers = jedis.smembers("orders");
for (String order : smembers) {
System.out.println(order);
}
jedis.srem("orders", "order02");
```

2.6 Jedis-API: Hash

```
jedis.hset("hash1","userName","lisi");
System.out.println(jedis.hget("hash1","userName"));
Map<String,String> map = new HashMap<String,String>();
map.put("telphone","13810169999");
map.put("address","atguigu");
map.put("email","abc@163.com");
jedis.hmset("hash2",map);
List<String> result = jedis.hmget("hash2", "telphone","email");
for (String element : result) {
System.out.println(element);
}
```

2.7 Jedis-API: Zset

```
jedis.zadd("zset01", 100d, "z3");
jedis.zadd("zset01", 90d, "l4");
jedis.zadd("zset01", 80d, "w5");
jedis.zadd("zset01", 70d, "z6");
Set<String> zrange = jedis.zrange("zset01", 0, -1);
for (String e : zrange) {
System.out.println(e);
}
```

3 Redis 与 SpringBoot的整合

3.1 引入依赖

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-redis</artifacted>
</dependency>
```

```
<!-- spring2.X集成redis所需common-pool2 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-pool2<artifacted>
    <version>2.6.0</verison>
</dependency>
```

3.2 applciation.properties 配置redis

```
# redis服务器地址
spring.redis.host=192.168.140.136
# redis服务器连接端口
spring.redis.port=6379
# redis数据库索引（默认为0）
spring.redis.database=0
# 连接超时时间（毫秒）、
spring.redis.timeout=1800000
# 连接池最大连接数（负数代表没有限制）
spring.redis.lettuce.pool.max-active-=20
# 最大阻塞等待时间（负数表示没有限制）
spring.redis.lettuce.pool.max-wait=-1
# 连接池中的最大空闲连接
spring.redis.lettuce.pool.max-idle=5
# 连接池中的最小空闲连接
spring.redis.lettuce.pool.min-idle=0
```

3.3 添加redis配置类

```
@EnableCaching
@Configuration
public class RedisConfig extends CachingConfigurerSupport {
    @Bean
    public RedisTemplate<String, Object> redisTemplate(RedisConnectionFactory factory) {
        RedisTemplate<String, Object> template = new RedisTemplate<>();
    RedisSerializer<String> redisSerializer = new StringRedisSerializer();
    Jackson2JsonRedisSerializer jackson2JsonRedisSerializer = new
        Jackson2JsonRedisSerializer(Object.class);
        ObjectMapper om = new ObjectMapper();
        om.setVisibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY);
        om.enableDefaultTyping(ObjectMapper.DefaultTyping.NON_FINAL);
    jackson2JsonRedisSerializer.setObjectMapper(om);
        template.setConnectionFactory(factory);
    // key序列化方式
    template.setKeySerializer(redisSerializer);
        // value序列化
    template.setValueSerializer(jackson2JsonRedisSerializer);
    // value hashmap序列化
    template.setHashValueSerializer(jackson2JsonRedisSerializer);
    return template;
    }
```

```
    @Bean
    public CacheManager cacheManager(RedisConnectionFactory factory) {
        RedisSerializer<String> redisSerializer = new StringRedisSerializer();
        Jackson2JsonRedisSerializer jackson2JsonRedisSerializer = new
        Jackson2JsonRedisSerializer(Object.class);
        //解决查询缓存转换异常的问题
        ObjectMapper om = new ObjectMapper();
        om.setVisibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY);
        om.enableDefaultTyping(ObjectMapper.DefaultTyping.NON_FINAL);
        jackson2JsonRedisSerializer.setObjectMapper(om);
        // 配置序列化（解决乱码的问题）,过期时间600秒
        RedisCacheConfiguration config = RedisCacheConfiguration.defaultCacheConfig()
                                        .entryTtl(Duration.ofSeconds(600))
     .serializeKeysWith(RedisSerializationContext.SerializationPair.fromSerializer(redisSerializer))

     .serializeValuesWith(RedisSerializationContext.SerializationPair.fromSerializer(
         jackson2JsonRedisSerializer))
      .disableCachingNullValues();
           RedisCacheManager cacheManager = RedisCacheManager.builder(factory)
                                        .cacheDefaults(config)
                                        .build();
         return cacheManager;
    }
}
```

3.4 测试一下

```
@RestController
@RequestMappiing("/redisTest")
public class RedisTestController {
    @Autowired
    private RedisTemplate redisTemplte;

    @GetMapping
    public Srting testRedis() {
        // 设置值到redis
        redisTemplate.opsForValue().set("name","lucy");
        // 从redis获取值
        String name = (String) redisTemplate.opsForvalue().get("name");
        return name;
    }
}
```