

SpringBoot操作Redis常用数据类型

1. 导入依赖 pom

<dependency>

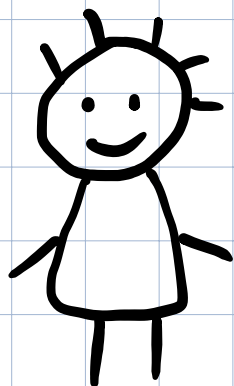
<groupId> org.springframework.boot </groupId>

<artifactId> spring-boot-starter-data-redis </artifactId>

</dependency>

2. redisTemplate 问题

```
@Bean
public CacheManager cacheManager(RedisTemplate redisTemplate) {
    RedisCacheManager rcm = new RedisCacheManager(redisTemplate);
    return rcm;
}
```



源码 springboot 2.x 中无了此 RedisTemplate 传参构造方法, 需改变一下此方法

```
@Bean
public CacheManager cacheManager(RedisConnectionFactory connectionFactory) {
    RedisCacheManager rcm = RedisCacheManager.create(connectionFactory);
    //设置缓存过期时间
    //rcm.setDefaultExpiration(60); //秒
    return rcm;
}
```

参考资料

<https://blog.csdn.net/u013317158/article/details/81285564>

3. 安装 Redis

① brew install redis → 安装 Homebrew

→ 一直报错 问题: 128

```
bogon:~ huawuque$ brew install redis
==> Tapping homebrew/cask
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-cask'...
fatal: unable to access 'https://github.com/Homebrew/homebrew-cask/': LibreSSL S
SL_connect: Connection reset by peer in connection to github.com:443
Error: Failure while executing; `git clone https://github.com/Homebrew/homebrew-
cask /usr/local/Homebrew/Library/Taps/homebrew/homebrew-cask --origin=origin --t
emplate=` exited with 128.
```

② 官网下载

解压后将文件放入 /usr/local 文件中 参考资料

<https://www.jianshu.com/p/3bdfda703552>

4. Redis 和 mysql 是同时存在吗?

~~好像~~ 不行, 创建项目勾选后会报错, 也有可能因为没有配置数据库。

5. 配置 Junit

6. Unable to connect to Redis;

1. 查看是否启动 Redis 服务器

a. 先停止 `redis-cli shutdown`

b. 开启 `redis-server` port: 6379 pid: 3503

2. 配置文件中 `spring.redis.timeout` 设置不可为 0

T. 创建 ReaderVo

name address age (Date) date sex (double) account (boolean) ifUse

J. TestTwo

`RedisTemplate` `ValueOperations` `HashOperations` `ListOperations` `SetOperations` `ZSetOperations`

Redis 模板

ps: redis 5 种数据类型: `String`. `List`. `Hash`. `Set`. `SortedSet`

▶ 1. 字符串 ○

- ▶ 1. set(set key value) 📄
- ▶ 2. get(get key)
- ▶ 3. setnx(setnx key value) 📄
- ▶ 4. setrange(setrange key startIndex vlaue)替换字符串 📄
- ▶ 5. mset(mset key1 value1 key2 value2 ...)批量设置键值对
- ▶ 6. msetnx(msetnx key1 value1 key2 value2...)如果key已存在那么设置失败
- ▶ 7. getset(getset key newValue)获取key的值然后设置新的值
- ▶ 8. getrange(getrange key startIndex endIndex)获取数据 📄
- ▶ 9. mget(mget key1 key2 key3...)批量获取
- ▶ 10. incr(incr key)自增1
- ▶ 11. incrby(incrby key num)指定增减的数量
- ▶ 12. decr(decr key)自减1
- ▶ 13. decrby(decrby key num)指定自增减的数量
- ▶ 14. append(append key value)给指定的字符串追加value的值
- ▶ 15. strlen(strlen key)获取指定的key对应的值得长度

▶ 2. List类型(类似于Java中的List) 📄 ○

- ▶ 1. lpush(lpush key value1 value2 ...)往list集合中压入元素
- ▶ 2. linsert(linsert key before/after value newValue)在指定的元素前或者后插入元素
- ▶ 3. lset(lset key index newValue)设置指定下标的值
- ▶ 4. lrem(lrem key count value)删除count个于value相同的元素,count>0从开始位置进行删除,count<0从末尾开始删除,count=0删除所有的
- ▶ 5. ltrim(ltrim key startIndex endIndex)删除指定范围内以外的元素,保留指定范围内的元素
- ▶ 6. lpop(lpop key)从list的头部删除元素
- ▶ 7. lindex(lindex key index)返回指定索引处的元素
- ▶ 8. llen(llen key)返回列表的长度
- ▶ 9. rpush(rpush key value)从末尾压入元素
- ▶ 10. rpop(rpop key)从末尾删除元素
- ▶ 11. rpoplpush(rpoplpush key1 key2)从key1链表中弹出最后一个元素然后压入到key2链表中

1. hset(hset key field value) 给指定的key添加key-value元素

2. hget(hget key field)获取指定的key中 field字段的值

3. hsetnx(hsetnx key field value)如果key不存在进行插入,如果key和field都存在不进行插入

4. hexists(hexists key field)判断指定的key中是否存在field这个字段

5. hlen(hlen key) 获取指定的key中字段的数量

6. hdel(hdel key field1 field2...)删除指定的key中的指定的字段的和对应的值

▶ 3. Hash(类似于Java中的Map) 📄 ○

7. hincrby(hincrby key field count)给指定的key的field的字段添加或者减去count这个值

8. hgetall(hgetall key)获取key中所有的键值对,返回的是一个键一个值

9. hkeys(hkeys key)获取指定的key中所有的字段

10. hmget(hmget key field1 field2...)获取指定的key中指定字段的值

11. hmset(hmset key field1 value1 field2 value2...)同时设置多个键值对数据

12. hvals(hvals key)获取指定的key中所有的value

13. hincrbyfloat(hincrbyfloat key field value)给指定的字段添加浮点数的值

1. sadd(sadd key member1 member2...)添加元素

2. scard(scard key)获取成员的数量

3. sismember(sismember key member)判断是否存在member这个成员

4. smembers(smembers key)获取所有的成员

5. spop(spop key)随机弹出一个成员

6. srandmember(srandmember key [count])随机获取一个或者多个成员

4. Set(类似于Java中的Set) 📄 ○

7. srem(srem key member1 member2 ...)删除一个或者多个成员,如果成员不存在则忽略

8. smove(smove source destination member)移动一个成员到指定的set中

9. sdiff(sdiff first-key key1 key2...)返回 给定集合之间的差集。不存在的集合 key 将视为空集

10. sdiffstore(sdiffstore destination key1 key2 ...)把获取到的差集保存到目标set中

11. sinter(sinter key1 key2...)获取交集

12. sinterstore(sinterstore destination key1 key2...)把获取到的交集存储到目标set中

13. sunion(sunion key1 key2...)获取并集

14. sunionstore(sunionstore destination key1 key2...)把获取到的并集存储到目标set中

5. SortedSet(和Set类似但是可以排序)

1. `zadd(zadd key score1 member1 score2 member2...)`添加成员
2. `zcard(zcard key)`计算元素个数
3. `zincrby(zincrby key number member)`给指定的member的分数添加或者减去 number这个值
4. `zcount(zcount key min max)`获取分数在min和max之间的成员和数量; 默认是闭区间; 想不包含可以: (min (max
5. `zrange(zrange key start stop [WITHSCORES])`返回指定排名之间的成员(结果是分数由低到高)
6. `zrevrange(zrevrange key start stop [WITHSCORES])`返回指定排名之间的成员(结果是分数由高到低)
7. `zrangebyscore(zrangebyscore key min max [withscores] [limit offset count])`根据分数的范围获取成员(按照分数: 从低到高)
8. `zrevrangebyscore(zrevrangebyscore key max min [withscores] [limit offset count])`根据分数的范围获取成员(从高到低)
9. `zrank(zrank key member)`返回一个成员的排名(从低到高的顺序)
10. `zrevrank(zrevrank key member)`返回一个成员的排名(从高到低)
11. `zscore(zscore key member)`获取一个成员的分数
12. `zrem(zrem key member1 member2...)`删除指定的成员
13. `zremrangebyrank(zremrangebyrank key start stop)`根据排名进行删除
14. `zremrangebyscore(zremrangebyscore key min max)`根据分数的范围进行删除

3. 对key的通用操作

- ▶ 1. `keys(keyspattern)`根据指定的规则返回符合条件的key
- ▶ 2. `del(del key1 key2...)`删除指定的key
- ▶ 3. `exists(exists key)`判断是否存在指定的key
4. `move(move key db)`将指定的key移入到指定的数据库中
5. `rename(rename key newkey)`对key进行重命名
6. `renamenx(renamenx key newkey)`仅当 newkey 不存在时, 将 key 改名为 newkey
- ▶ 7. `type(type key)`返回key的类型
- ▶ 8. `expire(expire key second)`给指定的key设置失效时间
9. `expireat(expireat key timestamp)`以时间戳的形式设置key的失效时间
10. `pexpireat(pexpireat key timestamp)`以毫秒为单位设置key的失效时间
- ▶ 11. `persist(persist key)`移除key的失效时间
12. `ttd(ttd key)`以秒为单位返回key的剩余时间(返回-2表示key不存在, 返回-1表示永远不过时)
13. `pttd(pttd key)`一毫秒为单位返回key的失效时间
14. `randomkey`: 随机返回一个key
15. `dump(dump key)`序列化给定 key

9. java Date变量

```
Date date = new Date()
```