# **[Springboot+Redis 配置和使用](https://www.cnblogs.com/-flq/p/9068605.html)**

**pom.xml 引入redis 开启缓存**

<!-- cache -->

<dependency>

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

<artifactId>**spring-boot-starter-cache**</artifactId>

</dependency>

<!-- redis -->

<dependency>

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

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

</dependency>

**application.properties 配置文件**

# Redis数据库索引（默认为0）

spring.redis.database**=**0

# Redis服务器地址

spring.redis.host**=**localhost

# Redis服务器连接端口

spring.redis.port**=**6379

# Redis服务器连接密码（默认为空）

spring.redis.password**=**

# 连接池最大连接数（使用负值表示没有限制）

spring.redis.pool.max-active**=**8

# 连接池最大阻塞等待时间（使用负值表示没有限制）

spring.redis.pool.max-wait**=**-1

# 连接池中的最大空闲连接

spring.redis.pool.max-idle**=**8

# 连接池中的最小空闲连接

spring.redis.pool.min-idle**=**0

# 连接超时时间（毫秒）

spring.redis.timeout**=**0

**RedisService**

**import** org**.**springframework**.**beans**.**factory**.**annotation**.**Autowired**;**

**import** org**.**springframework**.**data**.**redis**.**core**.\*;**

**import** org**.**springframework**.**stereotype**.**Service**;**

**import** java**.**io**.**Serializable**;**

**import** java**.**util**.**List**;**

**import** java**.**util**.**Set**;**

**import** java**.**util**.**concurrent**.**TimeUnit**;**

/\*\*

\* Created by Administrator on 2017/12/1.

\*/

@Service

public class RedisService **{**

@Autowired

private RedisTemplate redisTemplate**;**

/\*\*

\* 写入缓存

\* **@param** key

\* **@param** value

\* **@return**

\*/

public boolean set**(**final String key**,** Object value**)** **{**

boolean result **=** **false;**

**try** **{**

ValueOperations**<**Serializable**,** Object**>** operations **=** redisTemplate**.**opsForValue**();**

operations**.**set**(**key**,** value**);**

result **=** **true;**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**return** result**;**

**}**

/\*\*

\* 写入缓存设置时效时间

\* **@param** key

\* **@param** value

\* **@return**

\*/

public boolean set**(**final String key**,** Object value**,** Long expireTime**)** **{**

boolean result **=** **false;**

**try** **{**

ValueOperations**<**Serializable**,** Object**>** operations **=** redisTemplate**.**opsForValue**();**

operations**.**set**(**key**,** value**);**

redisTemplate**.**expire**(**key**,** expireTime**,** TimeUnit**.**SECONDS**);**

result **=** **true;**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**return** result**;**

**}**

/\*\*

\* 批量删除对应的value

\* **@param** keys

\*/

public void remove**(**final String**...** keys**)** **{**

**for** **(**String key **:** keys**)** **{**

remove**(**key**);**

**}**

**}**

/\*\*

\* 批量删除key

\* **@param** pattern

\*/

public void removePattern**(**final String pattern**)** **{**

Set**<**Serializable**>** keys **=** redisTemplate**.**keys**(**pattern**);**

**if** **(**keys**.**size**()** **>** 0**)**

redisTemplate**.**delete**(**keys**);**

**}**

/\*\*

\* 删除对应的value

\* **@param** key

\*/

public void remove**(**final String key**)** **{**

**if** **(**exists**(**key**))** **{**

redisTemplate**.**delete**(**key**);**

**}**

**}**

/\*\*

\* 判断缓存中是否有对应的value

\* **@param** key

\* **@return**

\*/

public boolean exists**(**final String key**)** **{**

**return** redisTemplate**.**hasKey**(**key**);**

**}**

/\*\*

\* 读取缓存

\* **@param** key

\* **@return**

\*/

public Object get**(**final String key**)** **{**

Object result **=** **null;**

ValueOperations**<**Serializable**,** Object**>** operations **=** redisTemplate**.**opsForValue**();**

result **=** operations**.**get**(**key**);**

**return** result**;**

**}**

/\*\*

\* 哈希 添加

\* **@param** key

\* **@param** hashKey

\* **@param** value

\*/

public void hmSet**(**String key**,** Object hashKey**,** Object value**){**

HashOperations**<**String**,** Object**,** Object**>** hash **=** redisTemplate**.**opsForHash**();**

hash**.**put**(**key**,**hashKey**,**value**);**

**}**

/\*\*

\* 哈希获取数据

\* **@param** key

\* **@param** hashKey

\* **@return**

\*/

public Object hmGet**(**String key**,** Object hashKey**){**

HashOperations**<**String**,** Object**,** Object**>** hash **=** redisTemplate**.**opsForHash**();**

**return** hash**.**get**(**key**,**hashKey**);**

**}**

/\*\*

\* 列表添加

\* **@param** k

\* **@param** v

\*/

public void lPush**(**String k**,**Object v**){**

ListOperations**<**String**,** Object**>** list **=** redisTemplate**.**opsForList**();**

list**.**rightPush**(**k**,**v**);**

**}**

/\*\*

\* 列表获取

\* **@param** k

\* **@param** l

\* **@param** l1

\* **@return**

\*/

public List**<**Object**>** lRange**(**String k**,** long l**,** long l1**){**

ListOperations**<**String**,** Object**>** list **=** redisTemplate**.**opsForList**();**

**return** list**.**range**(**k**,**l**,**l1**);**

**}**

/\*\*

\* 集合添加

\* **@param** key

\* **@param** value

\*/

public void add**(**String key**,**Object value**){**

SetOperations**<**String**,** Object**>** set **=** redisTemplate**.**opsForSet**();**

set**.**add**(**key**,**value**);**

**}**

/\*\*

\* 集合获取

\* **@param** key

\* **@return**

\*/

public Set**<**Object**>** setMembers**(**String key**){**

SetOperations**<**String**,** Object**>** set **=** redisTemplate**.**opsForSet**();**

**return** set**.**members**(**key**);**

**}**

/\*\*

\* 有序集合添加

\* **@param** key

\* **@param** value

\* **@param** scoure

\*/

public void zAdd**(**String key**,**Object value**,**double scoure**){**

ZSetOperations**<**String**,** Object**>** zset **=** redisTemplate**.**opsForZSet**();**

zset**.**add**(**key**,**value**,**scoure**);**

**}**

/\*\*

\* 有序集合获取

\* **@param** key

\* **@param** scoure

\* **@param** scoure1

\* **@return**

\*/

public Set**<**Object**>** rangeByScore**(**String key**,**double scoure**,**double scoure1**){**

ZSetOperations**<**String**,** Object**>** zset **=** redisTemplate**.**opsForZSet**();**

**return** zset**.**rangeByScore**(**key**,** scoure**,** scoure1**);**

**}**

**}**

**测试用的Controller**

**import** com**.**example**.**service**.**RedisService**;**

**import** org**.**springframework**.**beans**.**factory**.**annotation**.**Autowired**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RequestMapping**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RequestMethod**;**

**import** org**.**springframework**.**web**.**bind**.**annotation**.**RestController**;**

@RestController

public class DemoController **{**

@Autowired

private RedisService redisService **;**

@RequestMapping**(**value **=** "/test"**,**method **=** RequestMethod**.**POST**)**

public void demoTest**(){**

redisService**.**set**(**"1"**,**"value22222"**);**

**}**

**}**