JedisCluster + redis 3.2.5集群

开发环境

jedis 2.9.0

redis 3.2.5

springboot 1.3.6.RELEASE

注意事项：本文搭建的redis配置了密码，所以jedis客户端要使用比较新的版本，老版本不支持密码，需要修改源码，比较麻烦。

环境搭建

1. 引入jedis
2. <dependency>
3. <groupId>redis.clients</groupId>
4. <artifactId>jedis</artifactId>
5. <version>2.9.0</version>

</dependency>

1. 注入JedisClusterbean

import com.le.luffi.media.common.Constant;

import org.apache.commons.pool2.impl.GenericObjectPoolConfig;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import redis.clients.jedis.HostAndPort;

import redis.clients.jedis.JedisCluster;

import java.util.HashSet;

import java.util.Set;

@Configuration

public class JedisClusterConfig {

@Bean

public JedisCluster getJedisCluster() {

String[] serverArray = Constant.REDIS\_CACHE\_CLUSTER\_NODES.split(",");

Set<HostAndPort> nodes = new HashSet<>();

for (String ipPort : serverArray) {

String[] ipPortPair = ipPort.split(":");

nodes.add(new HostAndPort(ipPortPair[0].trim(), Integer.valueOf(ipPortPair[1].trim())));

}

JedisCluster jedisCluster = new JedisCluster(nodes, Constant.REDIS\_CACHE\_COMMANDTIMEOUT,

Constant.REDIS\_CACHE\_SOTIMEOUT, Constant.REDIS\_CACHE\_MAXATTEMPTS, Constant.REDIS\_CACHE\_CLUSTER\_PASSWORD, new GenericObjectPoolConfig());

return jedisCluster;

}

}

1. 创建JedisTemplate模板

import com.le.luffi.media.common.Constant;

import org.apache.commons.lang3.StringUtils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Component;

import redis.clients.jedis.JedisCluster;

@Component

public class JedisTemplate {

private static final Logger LOGGER = LoggerFactory.getLogger(JedisTemplate.class);

@Autowired

private JedisCluster jedisCluster;

private static final String KEY\_SPLIT = ":"; //用于隔开缓存前缀与缓存键值

/\*\*

\* 设置缓存

\* @param prefix 缓存前缀（用于区分缓存，防止缓存键值重复）

\* @param key 缓存key

\* @param value 缓存value

\*/

public void set(String prefix, String key, String value) {

if (StringUtils.isBlank(prefix)) throw new IllegalArgumentException("prefix must not null!");

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

jedisCluster.set(prefix + KEY\_SPLIT + key, value);

LOGGER.debug("RedisUtil:set cache key={},value={}", prefix + KEY\_SPLIT + key, value);

}

/\*\*

\* 设置缓存，并且自己指定过期时间

\* @param expireTime 过期时间

\*/

public void setWithExpireTime(String prefix, String key, String value, int expireTime) {

if (StringUtils.isBlank(prefix)) throw new IllegalArgumentException("prefix must not null!");

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

jedisCluster.setex(prefix + KEY\_SPLIT + key, expireTime, value);

LOGGER.debug("RedisUtil:setWithExpireTime cache key={},value={},expireTime={}", prefix + KEY\_SPLIT + key, value, expireTime);

}

/\*\*

\* 设置缓存，并且由配置文件指定过期时间

\*/

public void setWithExpireTime(String prefix, String key, String value) {

if (StringUtils.isBlank(prefix)) throw new IllegalArgumentException("prefix must not null!");

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

// int EXPIRE\_SECONDS = redisProperties.getExpireSeconds();

jedisCluster.setex(prefix + KEY\_SPLIT + key, Constant.REDIS\_CACHE\_EXPIRESECONDS, value);

LOGGER.debug("RedisUtil:setWithExpireTime cache key={},value={},expireTime={}", prefix + KEY\_SPLIT + key, value, Constant.REDIS\_CACHE\_EXPIRESECONDS);

}

/\*\*

\* 获取指定key的缓存

\*

\* @param prefix

\* @param key

\*/

public String get(String prefix, String key) {

if (StringUtils.isBlank(prefix)) throw new IllegalArgumentException("prefix must not null!");

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

String value = jedisCluster.get(prefix + KEY\_SPLIT + key);

LOGGER.debug("RedisUtil:get cache key={},value={}", prefix + KEY\_SPLIT + key, value);

return value;

}

/\*\*

\* 删除指定key的缓存

\*

\* @param prefix

\* @param key

\*/

public void deleteWithPrefix(String prefix, String key) {

if (StringUtils.isBlank(prefix)) throw new IllegalArgumentException("prefix must not null!");

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

jedisCluster.del(prefix + KEY\_SPLIT + key);

LOGGER.debug("RedisUtil:delete cache key={}", prefix + KEY\_SPLIT + key);

}

public void delete(String key) {

if (StringUtils.isBlank(key)) throw new IllegalArgumentException("key must not null!");

jedisCluster.del(key);

LOGGER.debug("RedisUtil:delete cache key={}", key);

}

}

4. controller简单测试

————————————————

版权声明：本文为CSDN博主「快乐崇拜234」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/liubenlong007/article/details/53689306

1. controller简单测试

import org.apache.commons.lang3.StringUtils;

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

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

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

@RestController

@RequestMapping("/user")

public class RedisController {

@Autowired

private JedisTemplate jedisTemplate;

@RequestMapping("/testJedisCluster")

public void testJedisCluster() {

String username = "username";

String value = jedisTemplate.get(RedisConstants.LUFFI\_LIVE\_PREFIX, username);

if (StringUtils.isBlank(value)) {

jedisTemplate.set(RedisConstants.LUFFI\_LIVE\_PREFIX, username, "lbl");

}

value = jedisTemplate.get(RedisConstants.LUFFI\_LIVE\_PREFIX, username);

System.out.println("value: " + value);

}

@RequestMapping("/deleteWithPrefix")

public void deleteWithPrefix() {

String username = "username";

jedisTemplate.deleteWithPrefix(RedisConstants.LUFFI\_LIVE\_PREFIX, username);

}

}

demo到此为止。