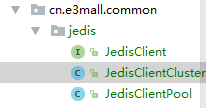
# 工具类(接口封装)

常用的操作redis的方法提取出一个接口，分别对应单机版和集群版创建两个实现类。



## 接口定义

**public interface** JedisClient {  
  
 String set(String key, String value);  
 String get(String key);  
 Boolean exists(String key);  
 Long expire(String key, **int** seconds);  
 Long ttl(String key);  
 Long incr(String key);  
 Long hset(String key, String field, String value);  
 String hget(String key, String field);  
 Long hdel(String key, String... field);  
 Boolean hexists(String key, String field);  
 List<String> hvals(String key);  
 Long del(String key);  
}

## 注意

注意：单机版和集群版不能共存，使用单机版时注释集群版的配置。使用集群版，把单机版注释。

# 单机版

## 实现类

**public class** JedisClientPool **implements** JedisClient {  
   
 **private** JedisPool **jedisPool**;  
  
 **public** JedisPool getJedisPool() {  
 **return jedisPool**;  
 }  
  
 **public void** setJedisPool(JedisPool jedisPool) {  
 **this**.**jedisPool** = jedisPool;  
 }  
  
 @Override  
 **public** String set(String key, String value) {  
 Jedis jedis = **jedisPool**.getResource();  
 String result = jedis.set(key, value);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** String get(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 String result = jedis.get(key);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Boolean exists(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 Boolean result = jedis.exists(key);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long expire(String key, **int** seconds) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.expire(key, seconds);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long ttl(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.ttl(key);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long incr(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.incr(key);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long hset(String key, String field, String value) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.hset(key, field, value);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** String hget(String key, String field) {  
 Jedis jedis = **jedisPool**.getResource();  
 String result = jedis.hget(key, field);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long hdel(String key, String... field) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.hdel(key, field);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Boolean hexists(String key, String field) {  
 Jedis jedis = **jedisPool**.getResource();  
 Boolean result = jedis.hexists(key, field);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** List<String> hvals(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 List<String> result = jedis.hvals(key);  
 jedis.close();  
 **return** result;  
 }  
  
 @Override  
 **public** Long del(String key) {  
 Jedis jedis = **jedisPool**.getResource();  
 Long result = jedis.del(key);  
 jedis.close();  
 **return** result;  
 }  
  
}

## applicationContext-redis.xml

配置：applicationContext-redis.xml

*<?***xml version="1.0" encoding="UTF-8"***?>*<**beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:context="http://www.springframework.org/schema/context" xmlns:p="http://www.springframework.org/schema/p"  
 xmlns:aop="http://www.springframework.org/schema/aop" xmlns:tx="http://www.springframework.org/schema/tx"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.2.xsd  
 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.2.xsd  
 http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.2.xsd  
 http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.2.xsd"**>  
   
 *<!-- 连接redis单机版 -->* <**bean id="jedisClientPool" class="cn.e3mall.common.jedis.JedisClientPool"**>  
 <**property name="jedisPool" ref="jedisPool"**></**property**>  
 </**bean**>  
 <**bean id="jedisPool" class="redis.clients.jedis.JedisPool"**>  
 <**constructor-arg name="host" value="192.168.25.101"**/>  
 <**constructor-arg name="port" value="6379"**/>  
 </**bean**>  
</**beans**>

# 集群版

## 实现类

**public class** JedisClientCluster **implements** JedisClient {  
   
 **private** JedisCluster **jedisCluster**;  
   
 **public** JedisCluster getJedisCluster() {  
 **return jedisCluster**;  
 }  
  
 **public void** setJedisCluster(JedisCluster jedisCluster) {  
 **this**.**jedisCluster** = jedisCluster;  
 }  
  
 @Override  
 **public** String set(String key, String value) {  
 **return jedisCluster**.set(key, value);  
 }  
  
 @Override  
 **public** String get(String key) {  
 **return jedisCluster**.get(key);  
 }  
  
 @Override  
 **public** Boolean exists(String key) {  
 **return jedisCluster**.exists(key);  
 }  
  
 @Override  
 **public** Long expire(String key, **int** seconds) {  
 **return jedisCluster**.expire(key, seconds);  
 }  
  
 @Override  
 **public** Long ttl(String key) {  
 **return jedisCluster**.ttl(key);  
 }  
  
 @Override  
 **public** Long incr(String key) {  
 **return jedisCluster**.incr(key);  
 }  
  
 @Override  
 **public** Long hset(String key, String field, String value) {  
 **return jedisCluster**.hset(key, field, value);  
 }  
  
 @Override  
 **public** String hget(String key, String field) {  
 **return jedisCluster**.hget(key, field);  
 }  
  
 @Override  
 **public** Long hdel(String key, String... field) {  
 **return jedisCluster**.hdel(key, field);  
 }  
  
 @Override  
 **public** Boolean hexists(String key, String field) {  
 **return jedisCluster**.hexists(key, field);  
 }  
  
 @Override  
 **public** List<String> hvals(String key) {  
 **return jedisCluster**.hvals(key);  
 }  
  
 @Override  
 **public** Long del(String key) {  
 **return jedisCluster**.del(key);  
 }  
  
}

## Spring的配置applicationContext-redis.xml：

*<!-- 连接redis集群 -->*<**bean id="jedisClientCluster" class="cn.e3mall.common.jedis.JedisClientCluster"**>  
 <**property name="jedisCluster" ref="jedisCluster"**/>  
</**bean**>  
<**bean id="jedisCluster" class="redis.clients.jedis.JedisCluster"**>  
 <**constructor-arg name="nodes"**>  
 <**set**>  
 <**bean class="redis.clients.jedis.HostAndPort"**>  
 <**constructor-arg name="host" value="192.168.25.101"**></**constructor-arg**>  
 <**constructor-arg name="port" value="6379"**></**constructor-arg**>  
 </**bean**>   
 <**bean class="redis.clients.jedis.HostAndPort"**>  
 <**constructor-arg name="host" value="192.168.25.101"**></**constructor-arg**>  
 <**constructor-arg name="port" value="6380"**></**constructor-arg**>  
 </**bean**>   
 <**bean class="redis.clients.jedis.HostAndPort"**>  
 <**constructor-arg name="host" value="192.168.25.101"**></**constructor-arg**>  
 <**constructor-arg name="port" value="6381"**></**constructor-arg**>  
 </**bean**>   
 <**bean class="redis.clients.jedis.HostAndPort"**>  
 <**constructor-arg name="host" value="192.168.25.101"**></**constructor-arg**>  
 <**constructor-arg name="port" value="6382"**></**constructor-arg**>  
 </**bean**>   
 </**set**>  
 </**constructor-arg**>  
</**bean**>

# 测试

单机版和集群版都一样，他们唯一区别就是spring配置不同

@Test  
**public void** testJedisClient() **throws** Exception {  
 *//初始化spring容器* ApplicationContext applicationContext = **new** ClassPathXmlApplicationContext(**"classpath:spring/applicationContext-redis.xml"**);  
 *//从容器中获得JedisClient对象* JedisClient jedisClient = applicationContext.getBean(JedisClient.**class**);  
 jedisClient.set(**"mytest"**, **"jedisClient"**);  
 String string = jedisClient.get(**"mytest"**);  
 System.***out***.println(string);  
}