

Spring 整合 Redis

作者: [Single](#)

原文链接: <https://ld246.com/article/1498719977727>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

接口

```
package com.util.redis;
/**
 * Redis工具接口_
 */
public interface RedisUtils {
    /**
     * 保存_
     * @param key
     *      键
     * @param value
     *      值
     */
    public void set(String key, String value);
    /**
     * 保存并设置存活时间_
     * @param key
     *      键
     * @param value
     *      值
     * @param seconds
     *      存活时间，单位为秒
     */
    public void set(String key, String value, Integer seconds);
    /**
     * 根据key查询_
     * @param key
     *      键
     * @return 值_ /
     */
    public String get(String key);
    /**
     * 删除_
     * @param key
     *      键
     */
    public void del(String key);
    /**
     * 根据key设置存活时间_
     * @param key
     *      键
     * @param seconds
     *      存活时间，单位为秒
     */
    public void expire(String key, Integer seconds);
}

##### 单机版实现类

package com.util.redis.impl;
import org.springframework.beans.factory.annotation.Autowired;
import com.util.redis.RedisUtils;
```

```

import redis.clients.jedis.Jedis;
import redis.clients.jedis.JedisPool;
/**
    _Redis连接池工具类_
    */
public class RedisPool implements RedisUtils {
    @Autowired
    private JedisPool jedisPool;
    @Override
    public void set(String key, String value) {
        Jedis jedis = this.getJedis();
        jedis.set(key, value);
        this.releaseJedis(jedis);
    }
    @Override
    public void set(String key, String value, Integer seconds) {
        Jedis jedis = this.getJedis();
        jedis.set(key, value);
        jedis.expire(key, seconds);
        this.releaseJedis(jedis);
    }
    @Override
    public String get(String key) {
        Jedis jedis = this.getJedis();
        String result = jedis.get(key);
        this.releaseJedis(jedis);
        return result;
    }
    @Override
    public void del(String key) {
        Jedis jedis = this.getJedis();
        jedis.del(key);
        this.releaseJedis(jedis);
    }
    @Override
    public void expire(String key, Integer seconds) {
        Jedis jedis = this.getJedis();
        jedis.expire(key, seconds);
        this.releaseJedis(jedis);
    }
}
/**
    _获取Jedis连接_
    _@return Jedis连接_ /
    public Jedis getJedis() {
        return this.jedisPool.getResource();
    }
    / *
    _释放Jedis连接_
    _@param jedis_
    _jedis连接_ /
    public void releaseJedis(Jedis jedis) {
        jedis.close();
    }
}

```

集群版实现类

```
package com.util.redis.impl;
import org.springframework.beans.factory.annotation.Autowired;
import com.util.redis.RedisUtils;
import redis.clients.jedis.JedisCluster;
/**
 _Redis集群工具类_
 */
public class RedisCluster implements RedisUtils {
    @Autowired
    private JedisCluster jedisCluster;
    @Override
    public void set(String key, String value) {
        this.jedisCluster.set(key, value);
    }
    @Override
    public void set(String key, String value, Integer seconds) {
        this.jedisCluster.set(key, value);
        this.jedisCluster.expire(key, seconds);
    }
    @Override
    public String get(String key) {
        return this.jedisCluster.get(key);
    }
    @Override
    public void del(String key) {
        this.jedisCluster.del(key);
    }
    @Override
    public void expire(String key, Integer seconds) {
        this.jedisCluster.expire(key, seconds);
    }
}
/**
 _获取JedisCluster对象
 _可以直接使用它来进行redis操作
 _@return JedisCluster对象
 */
public JedisCluster getJedisCluster() {
    return jedisCluster;
}
}
```

redis.properties

```
#redis单机版信息
redis.host=192.168.37.131
redis.port=6379
#redis集群版信息
cluster.host1=192.168.37.131
cluster.port1=7001
cluster.host2=192.168.37.131
cluster.port2=7002
```

```
cluster.host3=192.168.37.131
cluster.port3=7003
cluster.host4=192.168.37.131
cluster.port4=7004
cluster.host5=192.168.37.131
cluster.port5=7005
cluster.host6=192.168.37.131
cluster.port6=7006
```

applicationContext-redis.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       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.xsd">
    <!-- 两种方案：使用jedis连接池(单机版)和jedis集群 -->
    <!-- 注意：如果在注入工具类的时候是按类型注入的话，那么不使用的工具类必须注释掉-->
    <!-- 配置 读取外部properties文件 -->
    <context:property-placeholder location="classpath:resource/*.properties" />
    <!-- 单机版 -->
    <!-- 配置jedis连接池 -->
    <bean class="redis.clients.jedis.JedisPool">
        <constructor-arg name="host" value="${redis.host}" />
        <constructor-arg name="port" value="${redis.port}" />
    </bean>
    <!-- 配置单机版工具类 -->
    <bean class="com.util.redis.impl.RedisPool" />
    <!-- 集群版 -->
    <!-- 配置jedis集群 -->
    <bean class="redis.clients.jedis.JedisCluster">
        <constructor-arg name="nodes">
            <set>
                <bean class="redis.clients.jedis.HostAndPort">
                    <constructor-arg name="host" value="${cluster.host1}" />
                    <constructor-arg name="port" value="${cluster.port1}" />
                </bean>
                <bean class="redis.clients.jedis.HostAndPort">
                    <constructor-arg name="host" value="${cluster.host2}" />
                    <constructor-arg name="port" value="${cluster.port2}" />
                </bean>
                <bean class="redis.clients.jedis.HostAndPort">
                    <constructor-arg name="host" value="${cluster.host3}" />
                    <constructor-arg name="port" value="${cluster.port3}" />
                </bean>
                <bean class="redis.clients.jedis.HostAndPort">
                    <constructor-arg name="host" value="${cluster.host4}" />
                    <constructor-arg name="port" value="${cluster.port4}" />
                </bean>
                <bean class="redis.clients.jedis.HostAndPort">
                    <constructor-arg name="host" value="${cluster.host5}" />
                    <constructor-arg name="port" value="${cluster.port5}" />
                </bean>
            </set>
        </constructor-arg>
    </bean>
```

```
        <bean class="redis.clients.jedis.HostAndPort">
            <constructor-arg name="host" value="${cluster.host6}" />
            <constructor-arg name="port" value="${cluster.port6}" />
        </bean>
    </set>
</constructor-arg>
</bean>
<!-- 配置集群版工具类 -->
<bean class="com.util.redis.impl.RedisCluster" />
</beans>
```