



链滴

# Redis 学习日志之 SpringBoot2.0+ 整合 Redis (基于 Redission)

作者: [hjljy](#)

原文链接: <https://ld246.com/article/1552533990699>

来源网站: [链滴](#)

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



海加尔金鹰 —— 如野草般一岁一枯荣

昨天完成了redis的简单安装[Redis学习日志之Linux下的安装](#)，今天就在项目当中简单整合使用一下

## 整合环境说明

JDK1.8版本

idea开发工具

springboot2.1.0版本

## 为什么选择Redisson?

springboot2.0之前的版本默认支持的是Jedis 但是在2.0后换成了Lettuce。

java连接redis最多的就是Jedis,Redisson,Lettuce这三种方式

简单说明介绍: [Redis的三个框架: Jedis,Redisson,Lettuce](#)

最后选择了Redisson, 因为Redisson的宗旨是促进使用者对Redis的关注分离, 从而让使用者能够精力更集中地放在处理业务逻辑上。

## 在POM当中引入对应JAR

根据[Redisson官方整合文档](#)进行引入和使用:

```
<!-- redisson -->
<dependency>
  <groupId>org.redisson</groupId>
  <artifactId>redisson-spring-boot-starter</artifactId>
  <version>3.10.4</version>
</dependency>
```

# 创建配置类RedissonConfig

```
package com.hjljy.blog.redission;

import org.redisson.Redisson;
import org.redisson.api.RedissonClient;
import org.redisson.config.Config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

/**
 * @Author: 海加尔金鹰
 * @Date: 2019年03月12日 17:47
 * @Description:
 */
@Configuration
public class RedissonConfig {

    @Bean
    public RedissonClient getRedisson(){
        Config config = new Config();
        //单机模式 依次设置redis地址和密码
        config.useSingleServer().
            setAddress("redis://IP地址:6379").
            setPassword("你的密码");
        return Redisson.create(config);
    }
}
```

这里只配置了最基本的参数，redis的配置方式有多种，配置参数也有很多，具体见[官方参数配置说明中文](#))

## 编写测试类进行测试

在进行测试之前，由于使用的是阿里云服务器，所以需要在安全组当中开放6379端口，不然连接不上。

```
@RunWith(SpringRunner.class)
@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.RANDOM_PORT)
public class BlogApplicationTests {

    @Resource
    private RedissonClient redissonClient;

    @Test
    public void RedissonTest(){
        RBucket<String> key = redissonClient.getBucket("newday");
        key.set("新的数据");
        System.out.println("获取到新存入的数据: "+key.get());
        // 获取字符串格式的数据
        RBucket<String> keyObj = redissonClient.getBucket("myname");
        String s = keyObj.get();
        System.out.println("获取到昨天存入的数据: "+s);
    }
}
```

```
}
```

测试结果：

获取到新存入的数据：新的数据

```
[ ERROR] [2019-03-14 10:37:37] [redisson-netty-1-9] org.redisson.client.handler.CommandDe  
oder [204] - Unable to decode data. channel: [id: 0x704603a9, L:/192.168.0.147:10879 - R:47.9  
.139.213/47.94.139.213:6379], reply: ReplayingDecoderByteBuf(ridx=9, widx=9), command: (G  
T), params: [myname]  
java.io.IOException: java.lang.NullPointerException
```

可以获取到项目当中新存入的数据，但是无法获取到昨天通过redis-cli存入的数据。

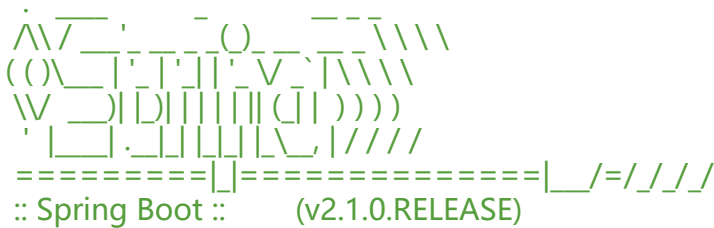
错误信息提示：无法解码数据，猜测可能是redis-cli存数据的编码和redission的编码不一致导致的。

最后网上找了很多的资料，找到了一个叫序列化策略的东西。大致情况就是说redis-cli存数据时的序列化策略是string，但是redission的默认序列化策略是Jackson JSON 编码 [Redisson官网配置参  
序列化说明](#)

## 修改配置类RedissonConfig如下

```
@Bean  
public RedissonClient getRedisson(){  
    Config config = new Config();  
    //单机模式 依次设置redis地址和密码  
    config.useSingleServer().  
        setAddress("redis://IP地址:6379").  
        setPassword("你的密码");  
    config.setCodec(new StringCodec());  
    return Redisson.create(config);  
}
```

最后结果如下：



获取到新存入的数据：新的数据

获取到昨天存入的数据：ycf

到这里简单的使用就完成了。

## 总结

- 1 RedissonClient 这个类很重要，可以看到数据的读取都是通过这个类进行的。
- 2 这么多种序列化方式，所以序列化可能是个坑，后面需要注意。
- 3 官方文档的说明很详细，没事可以多看看。

## 参考资料

[Redisson官方文档](#)

[springboot 整合redisson](#)