

SpringBoot2 中使用 Redis

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1534673123527>

来源网站: [链滴](#)

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

前言

简介

SpringBoot2中，使用Redis，官方提供了两个starter：

- spring-boot-starter-data-redis
- spring-boot-starter-data-redis-reactive

其实spring-boot-starter-data-redis-reactive中也仅依赖了spring-boot-starter-data-redis

第一个例子

由于SpringBoot的特点，所以上手是最容易的。

依赖

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.4.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
</dependency>
```

Controller

```
package com.example.testredis;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DataAccessException;
import org.springframework.data.redis.connection.RedisConnection;
import org.springframework.data.redis.core.RedisCallback;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.data.redis.core.StringRedisTemplate;
import org.springframework.data.redis.serializer.RedisSerializer;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

```

@RestController
public class TestController {

    private static final Logger logger = LoggerFactory.getLogger(TestController.class);

    @Autowired
    RedisTemplate redisTemplate;

    @Autowired
    StringRedisTemplate stringRedisTemplate;

    @RequestMapping("/set")
    public void set(String key, String value) {
        stringRedisTemplate.opsForValue().set(key, value);
    }

    @RequestMapping("/get")
    public String show(String key) {
        return stringRedisTemplate.opsForValue().get(key);
    }

    @RequestMapping("/set2")
    public void set2(String key, String value) {
        // 可选，自定义序列化方法
        stringRedisTemplate.execute(new RedisCallback<Boolean>() {
            @Override
            public Boolean doInRedis(RedisConnection redisConnection) throws DataAccessException
on {
                RedisSerializer<String> redisSerializer = redisTemplate.getStringSerializer();
                byte[] serializeKey = redisSerializer.serialize(key);
                byte[] serializeValue = redisSerializer.serialize(value);
                return redisConnection.setNX(serializeKey, serializeValue);
            }
        });
    }
}

```

执行

```
$ curl -s "localhost:8080/get?key=host"
note.abeffect.com
```

```
$ curl -s "localhost:8080/set?key=key&value=abeffect"
```

```
$ curl -s "localhost:8080/get?key=key"
abeffect
```

```
$ curl -s "localhost:8080/set2?key=key&value=abeffect"
```

```
$ curl -s "localhost:8080/get?key=key"
abeffect
```

参考

- [Spring Boot使用Spring Data Redis操作Redis \(单机/集群\)](#)
- [一起来学SpringBoot | 第九篇：整合Lettuce Redis](#): 有属性配置，序列化实体类，自定义Template，多线程测试
- [Spring Boot 2.0 新特性和发展方向](#): 2.0的新东西，包括但不限于mongodb-reactive, cassandra-reactive, spring-boot-starter-json, JdbcTemplate, JSON-B等
- [SpringBoot整合Redis及Redis工具类撰写](#): RedisTemplate的自动配置原理, 写自己的RedisTemplate, 写一个Redis工具类