Spring Data Redis工具类

叹雪飞花 2023-07-08 208 阅读4分钟

关注

分享一个在Spring Data Redis的基础上封装的工具类，本文章主要是记录，以后可能会更新。

java 复制代码

```java
package com.example.support;

import com.example.util.JsonUtils;
import jakarta.annotation.Resource;
import org.springframework.data.redis.core.HashOperations;
import org.springframework.data.redis.core.ListOperations;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.data.redis.core.ValueOperations;
import org.springframework.stereotype.Component;
import org.springframework.util.ObjectUtils;

import java.util.Arrays;
import java.util.Collection;
import java.util.Map;
import java.util.concurrent.TimeUnit;

/**
 * Redis操作类
 *
 * @param <V> value的类型
 * @author vains
 */
@Component
public class RedisOperator<V> {

    /**
     * 这里使用 @Resource 注解是因为在配置文件中注入ioc的泛型是<Object, Object>，所以类型匹配不上，
     * resource是会先根据名字去匹配的，所以使用Resource注解可以成功注入
     */
    @Resource
    private RedisTemplate<String, V> redisTemplate;

    @Resource
```

```
37      * 设置key的过期时间
38      *
39      * @param key     缓存key
40      * @param timeout 存活时间
41      * @param unit    时间单位
42      */
43     public void setExpire(String key, long timeout, TimeUnit unit) {
44         redisHashTemplate.expire(key, timeout, unit);
45     }
46
47     /**
48      * 根据key删除缓存
49      *
50      * @param keys 要删除的key，可变参数列表
51      * @return 删除的缓存数量
52      */
53     public Long delete(String... keys) {
54         if (ObjectUtils.isEmpty(keys)) {
55             return 0L;
56         }
57         return redisTemplate.delete(Arrays.asList(keys));
58     }
59
60     /**
61      * 存入值
62      *
63      * @param key   缓存中的key
64      * @param value 存入的value
65      */
66     public void set(String key, V value) {
67         valueOperations().set(key, value);
68     }
69
70     /**
71      * 根据key取值
72      *
73      * @param key 缓存中的key
74      * @return 返回键值对应缓存
75      */
76     public V get(String key) {
77         return valueOperations().get(key);
78     }
79
80     /**
81      * 设置键值并设置过期时间
82      *
```

```
86       * @param unit    过期时间的单位
87       */
88      public void set(String key, V value, long timeout, TimeUnit unit) {
89          valueOperations().set(key, value, timeout, unit);
90      }
91
92      /**
93       * 设置键值并设置过期时间（单位秒）
94       *
95       * @param key     键
96       * @param value   值
97       * @param timeout 过期时间,单位：秒
98       */
99      public void set(String key, V value, long timeout) {
100         this.set(key, value, timeout, TimeUnit.SECONDS);
101     }
102
103     /**
104      * 根据key获取缓存并删除缓存
105      *
106      * @param key 要获取缓存的key
107      * @return key对应的缓存
108      */
109     public V getAndDelete(String key) {
110         if (ObjectUtils.isEmpty(key)) {
111             return null;
112         }
113         V value = valueOperations().get(key);
114         this.delete(key);
115         return value;
116     }
117
118     /**
119      * 往hash类型的数据中存值
120      *
121      * @param key   缓存中的key
122      * @param field hash结构的key
123      * @param value 存入的value
124      */
125     public void setHash(String key, String field, V value) {
126         hashOperations().put(key, field, value);
127     }
128
129     /**
130      * 根据key取值
131      *
```

```
135     public Object getHash(String key, String field) {
136         return hashOperations().hasKey(key, field) ? hashOperations().get(key, field) : null;
137     }
138
139     /**
140      * 以hash格式存入redis
141      *
142      * @param key    缓存中的key
143      * @param value 存入的对象
144      */
145     public void setHashAll(String key, Object value) {
146         Map<String, Object> map = JsonUtils.objectCovertToObject(value, Map.class, String.class,
147         hashOperations().putAll(key, map);
148     }
149
150     /**
151      * 设置键值并设置过期时间
152      *
153      * @param key      键
154      * @param value    值
155      * @param timeout 过期时间
156      * @param unit     过期时间的单位
157      */
158     public void setHashAll(String key, Object value, long timeout, TimeUnit unit) {
159         this.setHashAll(key, value);
160         this.setExpire(key, timeout, unit);
161     }
162
163     /**
164      * 设置键值并设置过期时间（单位秒）
165      *
166      * @param key      键
167      * @param value    值
168      * @param timeout 过期时间,单位：秒
169      */
170     public void setHashAll(String key, Object value, long timeout) {
171         this.setHashAll(key, value, timeout, TimeUnit.SECONDS);
172     }
173
174     /**
175      * 从redis中获取hash类型数据
176      *
177      * @param key 缓存中的key
178      * @return redis 中hash数据
179      */
180     public Map<String, Object> getMapHashAll(String key) {
```

```
184     /**
185      * 根据指定clazz类型从redis中获取对应的实例
186      *
187      * @param key    缓存key
188      * @param clazz hash对应java类的class
189      * @param <T>    redis中hash对应的java类型
190      * @return clazz实例
191      */
192     public <T> T getHashAll(String key, Class<T> clazz) {
193         Map<String, Object> entries = hashOperations().entries(key);
194         if (ObjectUtils.isEmpty(entries)) {
195             return null;
196         }
197         return JsonUtils.objectCovertToObject(entries, clazz);
198     }
199
200     /**
201      * 根据key删除缓存
202      *
203      * @param key     要删除的key
204      * @param fields key对应的hash数据的键值(HashKey)，可变参数列表
205      * @return hash删除的属性数量
206      */
207     public Long deleteHashField(String key, String... fields) {
208         if (ObjectUtils.isEmpty(key) || ObjectUtils.isEmpty(fields)) {
209             return 0L;
210         }
211         return hashOperations().delete(key, (Object[]) fields);
212     }
213
214     /**
215      * 将value添加至key对应的列表中
216      *
217      * @param key    缓存key
218      * @param value 值
219      */
220     public void listPush(String key, V value) {
221         listOperations().rightPush(key, value);
222     }
223
224     /**
225      * 将value添加至key对应的列表中，并添加过期时间
226      *
227      * @param key      缓存key
228      * @param value    值
229      * @param timeout key的存活时间
```

```
233         listOperations().rightPush(key, value);
234         this.setExpire(key, timeout, unit);
235     }
236 
237     /**
238      * 将value添加至key对应的列表中，并添加过期时间
239      * 默认单位是秒(s)
240      *
241      * @param key     缓存key
242      * @param value   值
243      * @param timeout key的存活时间
244      */
245     public void listPush(String key, V value, long timeout) {
246         this.listPush(key, value, timeout, TimeUnit.SECONDS);
247     }
248 
249     /**
250      * 将传入的参数列表添加至key的列表中
251      *
252      * @param key    缓存key
253      * @param values 值列表
254      * @return 存入数据的长度
255      */
256     public Long listPushAll(String key, Collection<V> values) {
257         return listOperations().rightPushAll(key, values);
258     }
259 
260     /**
261      * 将传入的参数列表添加至key的列表中，并设置key的存活时间
262      *
263      * @param key     缓存key
264      * @param values  值列表
265      * @param timeout key的存活时间
266      * @param unit    时间单位
267      * @return 存入数据的长度
268      */
269     public Long listPushAll(String key, Collection<V> values, long timeout, TimeUnit unit) {
270         Long count = listOperations().rightPushAll(key, values);
271         this.setExpire(key, timeout, unit);
272         return count;
273     }
274 
275     /**
276      * 将传入的参数列表添加至key的列表中，并设置key的存活时间
277      *  默认单位是秒(s)
278      *
```

```
282      * @return 存入数据的长度
283      */
284     public Long listPushAll(String key, Collection<V> values, long timeout) {
285         return this.listPushAll(key, values, timeout, TimeUnit.SECONDS);
286     }
287 
288     /**
289      * 根据key获取list列表
290      *
291      * @param key 缓存key
292      * @return key对应的list列表
293      */
294     public Collection<V> getList(String key) {
295         Long size = listOperations().size(key);
296         if (size == null || size == 0) {
297             return null;
298         }
299         return listOperations().range(key, 0, (size - 1));
300     }
301 
302     /**
303      * value操作集
304      *
305      * @return ValueOperations
306      */
307     private ValueOperations<String, V> valueOperations() {
308         return redisTemplate.opsForValue();
309     }
310 
311     /**
312      * hash操作集
313      *
314      * @return ValueOperations
315      */
316     private HashOperations<String, String, Object> hashOperations() {
317         return redisHashTemplate.opsForHash();
318     }
319 
320     /**
321      * hash操作集
322      *
323      * @return ValueOperations
324      */
325     private ListOperations<String, V> listOperations() {
326         return redisTemplate.opsForList();
327     }
```

标签： Spring Boot Redis 话题： 我的技术写作成长之路

评论 0

抢首评，友善交流

0 / 1000 发送

暂无评论数据

相关推荐

Spring Authorization Server入门 (十四) 联合身份认证添加微信登录
1.2k阅读 · 5点赞

Spring Authorization Server常见问题解答(FAQ)
927阅读 · 1点赞

Spring Authorization Server优化篇：自定义UserDetailsService实现从数据库获取用户信息
1.1k阅读 · 5点赞

Spring Authorization Server优化篇：持久化JWKSource，解决重启后无法解析AccessToken问题
1.1k阅读 · 5点赞

精选内容

全网最硬核的源码分析之——线程池源码分析
梦尘啊 · 261阅读 · 2点赞

Java 学习路线：基础知识、数据类型、条件语句、函数、循环、异常处理、数据结构、面向对象编程、包、...
小万哥丶 · 151阅读 · 1点赞

初步探索Jenkins技术：持续集成与自动化部署的利器
牛哥说我不优雅 · 168阅读 · 3点赞

JVM工作原理与实战(二十九)：监控内存泄漏的工具
橘子青衫 · 157阅读 · 0点赞

JVM工作原理与实战(二十八)：内存溢出和内存泄漏
橘子青衫 · 152阅读 · 0点赞

为你推荐

Spring Authorization Server入门 (二) Spring Boot整合Spring Authorization Server
叹雪飞花 9月前 6.8k 31 86 Java

Spring Authorization Server入门 (十二) 实现授权码模式使用前后端分离的登录页面
叹雪飞花 8月前 4.8k 24 65 后端 Spring ... Spring

Spring Authorization Server入门 (十) 添加短信验证码方式登录
叹雪飞花 9月前 3.4k 20 9 Spring Spring ...

Spring Authorization Server入门 (八) Spring Boot引入Security OAuth2 Client对接认...
叹雪飞花 9月前 2.8k 13 43 Spring ... Spring

Spring Authorization Server入门 (十六) Spring Cloud Gateway对接认证服务
叹雪飞花 7月前 2.6k 17 44 Spring ... Spring ... 安全

Spring Authorization Server入门 (十三) 实现联合身份认证，集成Github与Gitee的OAu...
叹雪飞花 8月前 2.3k 13 51 Spring Spring ... 安全

Spring Authorization Server入门 (十一) 自定义grant_type(短信认证登录)获取token

Spring Authorization Server入门 (七) 登录添加图形验证码

叹雪飞花 9月前 2.9k 18 4 Spring ...

SpringBoot3.x最简集成SpringDoc-OpenApi

叹雪飞花 4月前 2.3k 16 评论 后端 Spring ... Java

Spring Authorization Server入门 (九) Spring Boot引入Resource Server对接认证服务

叹雪飞花 9月前 1.8k 13 8 Spring Spring ...

Spring Authorization Server优化篇：添加Redis缓存支持和统一响应类

叹雪飞花 8月前 1.7k 6 12 Spring Spring ... 安全

Spring Authorization Server入门 (十五) 分离授权确认与设备码校验页面

叹雪飞花 7月前 1.6k 14 8 Spring ... Spring Vue.js

Spring Authorization Server入门 (十九) 基于Redis的Token、客户端信息和授权确认信...

叹雪飞花 4月前 1.2k 8 19 Spring ... 后端 Redis

Spring Authorization Server入门 (二十) 实现二维码扫码登录

叹雪飞花 2月前 925 16 6 Spring ... Spring Java

Spring Authorization Server入门 (十七) Vue项目使用授权码模式对接认证服务

叹雪飞花 6月前 760 9 12 Vue.js 安全 Spring ...