# Redis缓存注解使用规范

版本号:1.0.0

**文档修改记录**

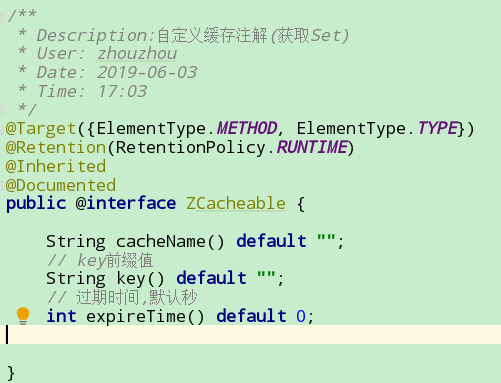
|  |  |  |  |
| --- | --- | --- | --- |
| **修改日期** | **修改人** | **修改说明** | **版本号** |
| 2019-06-04 | 周舟 | 自定义缓存直接的使用 | 1.0.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 注解说明

软件中一共使用到2个自定义注解,@ZCacheable, @ZCachePut都是方法级别的注解

### 1.1.1 @ZCacheable :主要是用来增强查询方法。

当缓存中有对应配置的缓存就会从缓存中获取值, 反之执行方法体的逻辑进行获取, 方法结束后会将方法返回值放入缓存。



里面有三个值:

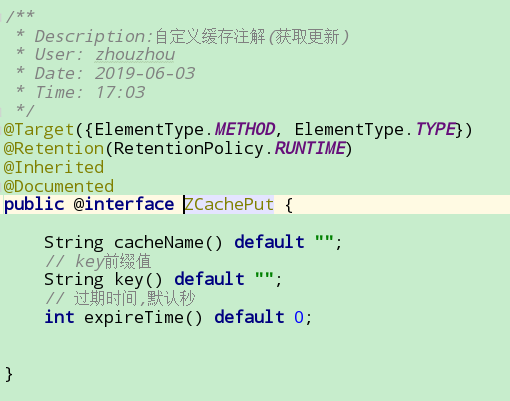
cacheName用来标注,缓存的名称(缓存的域)

Key用来标注缓存的key, 该值支持SPEL的语法

ExpireTime用来标注过期时间

### 1.1.2 @ZCachePut:主要是用来增强更新方法。

执行被该注解标注的方法后, 会将方法的返回值更新到缓存。

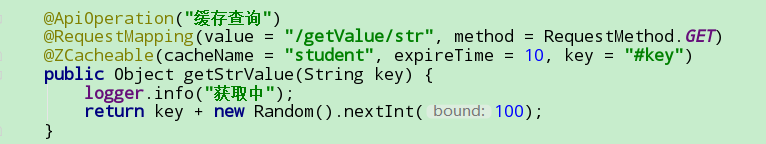


里面的字段和@ZCacheable 一样

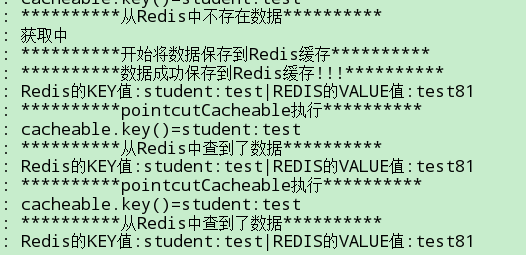
## 1.2注解的使用

### 1.2.1 @ZCacheable 的使用

直接在方法体上使用:如下图

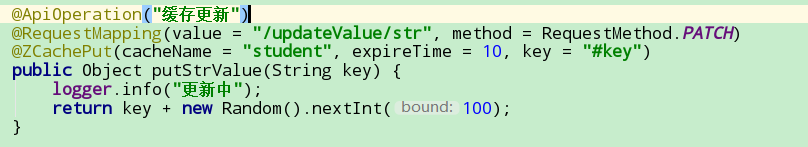


图示中, 缓存名(缓存域)为 student, 过期时间为10秒, key对应形参中的key,当我们启动项目不间断执行这个方法时候:通过日志可以看到, 我们的注解有效果了,第一次访问都时候进行查询, 后面几次都说直接从redis缓存中获取



### 1.2.2 @ZCachePut的使用

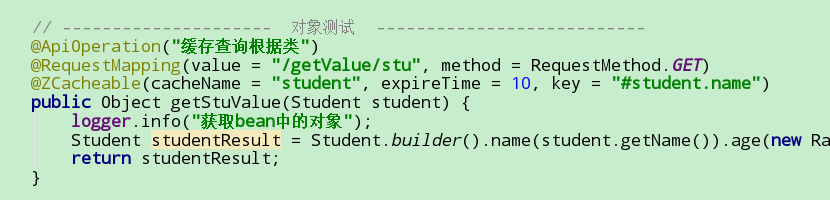
直接在方法体上使用:如下图



@ZCachePut都使用和@ZCacheable几乎一样,唯一功能性区别说, 每次方法体都会执行, 执行完更新缓存

## 1.3key值都SPEL高级使用

Key值满足spel语法 ,如下图使用到对象中都值调用



甚至字符串拼接, 判断等等.....