#### Spring Boot

##### 之Spring Cache- Redis实现

问题描述

Spring Boot 2.2.5

JDK 8

Spring Cache可以由多种Cache Provider提供功能支持，常用的有：

1. Jdk ConcurrentMap实现
2. Ehcache实现
3. Redis实现

这篇技术笔记记录了Redis实现。

Redis是分布式缓存的“老江湖”了。

Redis采用Sentinel（哨兵）机制进行投票，探测成员状态（Alive）。

解决

1. 安装Redis服务
2. 添加依赖

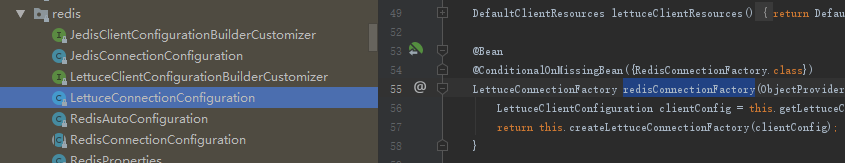
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-redis</artifactId>  
 <version>2.3.0.RELEASE</version>  
</dependency>

1. 添加配置类

@Configuration  
public class CacheConfig {  
// Redis Implementation  
 @Bean  
 public CacheManager cacheManager(RedisConnectionFactory lettuceConnectionFactory) {  
 RedisCacheConfiguration defaultCacheConfig = RedisCacheConfiguration.*defaultCacheConfig*();  
 // 设置缓存管理器管理的缓存的默认过期时间  
 defaultCacheConfig = defaultCacheConfig.entryTtl(Duration.*ofSeconds*(240))  
 // 设置 key为string序列化  
 .serializeKeysWith(RedisSerializationContext.SerializationPair.*fromSerializer*(new StringRedisSerializer()))  
 // 设置value为json序列化  
 .serializeValuesWith(RedisSerializationContext.SerializationPair.*fromSerializer*(new GenericJackson2JsonRedisSerializer()))  
 // 不缓存空值  
 .disableCachingNullValues();  
  
 Set<String> cacheNames = new HashSet<>();  
 cacheNames.add("products");  
  
 // 对每个缓存空间应用不同的配置  
 Map<String, RedisCacheConfiguration> configMap = new HashMap<>();  
 configMap.put("products", defaultCacheConfig.entryTtl(Duration.*ofSeconds*(180)));  
  
 RedisCacheManager cacheManager = RedisCacheManager.*builder*(lettuceConnectionFactory)  
 .cacheDefaults(defaultCacheConfig)  
 .initialCacheNames(cacheNames)  
 .withInitialCacheConfigurations(configMap)  
 .build();  
 return cacheManager;  
 }  
  
}

笔记

1. Java Bean Lettuce Connection Factory是由spring-boot-autoconfigure注入的：



□