#### Spring Boot

##### 之Spring Cache- 抽象层

问题描述

Spring Boot 2.2.5

JDK 8

日出东方，唯我不败。Spring框架作为大一统Java后端的万能框架，同样提供了缓存（Cache）的封装类。缓存的实现类分散在spring-context、spring-context-support两个Jar包中。

Spring5对Cache的实现进行了重构，主要由Cache Manager、Cache Resolver、AbstractValueAdaptingCache等Java类实现其功能。

Spring Cache的配置遵循“老三样”原则，即添加依赖、使能模块、添加注解。

解决

1. 添加依赖

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

1. 使能模块

@SpringBootApplication  
@MapperScan("com.bunny.spring.framework.dao.products.ibatis")  
@EnableCaching  
public class DAOApplication {  
 public static void main(String[] args) {  
 SpringApplication.*run*(DAOApplication.class);  
 }  
}

1. 添加注解

*/\*\*  
 \*  
 \* Spring Cache采用“懒加载”策略  
 \* 未实现的场景：  
 \* 由单次查询触发，取出（关联）表中所有数据项（生成项），载入缓存  
 \*  
 \*/*@CacheConfig(cacheNames = "products")  
@Service  
public class JdbcDaoService {  
  
 @Resource(name = "jdbcProductDao")  
 private ProductsDao jdbcProductDao;  
  
 @Cacheable  
 public Product queryById(Long id) throws Exception {  
 return jdbcProductDao.queryById(id);  
 }  
  
 @CachePut  
 public void update(Product product) throws Exception {  
 jdbcProductDao.update(product);  
 }  
  
 @CacheEvict(allEntries = true)  
 public Boolean evict() {  
 return true;  
 }  
}

笔记

1. Spring Cache功能尚不完备，不支持以下场景：
   1. 轻量级锁 Spring Cache使用重量级锁synchronized，同步机制不可配
   2. 一次触发，全部取出 即不遵循“懒加载”原则，由单次查询触发，加载表（关联表）中所有数据项（生成项）到缓存中

□