[**第十三篇：springboot集成spring cache**](http://blog.csdn.net/forezp/article/details/71023614)

参考：<http://blog.csdn.net/forezp/article/details/71023614>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/71023614>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

本文介绍如何在springboot中使用默认的spring cache，

声明式缓存

Spring 定义 CacheManager 和 Cache 接口用来统一不同的缓存技术。例如 JCache、 EhCache、 Hazelcast、 Guava、 Redis 等。在使用 Spring 集成 Cache 的时候，我们需要注册实现的 CacheManager 的 Bean。

Spring Boot 为我们自动配置了 JcacheCacheConfiguration、 EhCacheCacheConfiguration、HazelcastCacheConfiguration、GuavaCacheConfiguration、RedisCacheConfiguration、SimpleCacheConfiguration 等。

默认使用 ConcurrenMapCacheManager

在我们不使用其他第三方缓存依赖的时候，springboot自动采用ConcurrenMapCacheManager作为缓存管理器。

环境依赖

在pom文件引入spring-boot-starter-cache环境依赖：

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-cache</artifactId>

</dependency>

创建一个book数据访问层

先创建一个实体类

“`

public class Book {

private String isbn;

private String title;

public Book(String isbn, String title) {

this.isbn = isbn;

this.title = title;

}

….getter   
….setter

}   
“`

创建一个数据访问接口

public interface BookRepository {

Book getByIsbn(String isbn);

}

这个你可以写一个很复杂的数据查询操作，比如操作mysql、nosql等等。为了演示这个栗子，我只做了一下线程的延迟操作，当作是查询数据库的时间。

实现接口类：

@Component

public class SimpleBookRepository implements BookRepository {

@Override

public Book getByIsbn(String isbn) {

simulateSlowService();

return new Book(isbn, "Some book");

}

// Don't do this at home

private void simulateSlowService() {

try {

long time = 3000L;

Thread.sleep(time);

} catch (InterruptedException e) {

throw new IllegalStateException(e);

}

}

}

测试类

@Component

public class AppRunner implements CommandLineRunner {

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

private final BookRepository bookRepository;

public AppRunner(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

@Override

public void run(String... args) throws Exception {

logger.info(".... Fetching books");

logger.info("isbn-1234 -->" + bookRepository.getByIsbn("isbn-1234"));

logger.info("isbn-4567 -->" + bookRepository.getByIsbn("isbn-4567"));

logger.info("isbn-1234 -->" + bookRepository.getByIsbn("isbn-1234"));

logger.info("isbn-4567 -->" + bookRepository.getByIsbn("isbn-4567"));

logger.info("isbn-1234 -->" + bookRepository.getByIsbn("isbn-1234"));

logger.info("isbn-1234 -->" + bookRepository.getByIsbn("isbn-1234"));

}

}

启动程序，你会发现程序在控制台依次打印了：

2014-06-05 12:15:35.783 … : …. Fetching books

2014-06-05 12:15:40.783 … : isbn-1234 –> >Book{isbn=’isbn-1234’, title=’Some book’}

2014-06-05 12:15:43.784 … : isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}

2014-06-05 12:15:46.786 … : isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}

你会发现程序依次3s打印一行日志。这时还没开启缓存技术。

开启缓存技术

在程序的入口中加入@ EnableCaching开启缓存技术：

@SpringBootApplication

@EnableCaching

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

在需要缓存的地方加入@Cacheable注解，比如在getByIsbn（）方法上加入@Cacheable(“books”)，这个方法就开启了缓存策略，当缓存有这个数据的时候，会直接返回数据，不会等待去查询数据库。

@Component

public class SimpleBookRepository implements BookRepository {

@Override

@Cacheable("books")

public Book getByIsbn(String isbn) {

simulateSlowService();

return new Book(isbn, "Some book");

}

// Don't do this at home

private void simulateSlowService() {

try {

long time = 3000L;

Thread.sleep(time);

} catch (InterruptedException e) {

throw new IllegalStateException(e);

}

}

}

这时再启动程序，你会发现程序打印：

isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}   
2017-04-23 18:17:09.479 INFO 8054 — [ main] forezp.AppRunner : isbn-4567 –>Book{isbn=’isbn-4567’, title=’Some book’}   
2017-04-23 18:17:09.480 INFO 8054 — [ main] forezp.AppRunner : isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}   
2017-04-23 18:17:09.480 INFO 8054 — [ main] forezp.AppRunner : isbn-4567 –>Book{isbn=’isbn-4567’, title=’Some book’}   
2017-04-23 18:17:09.481 INFO 8054 — [ main] forezp.AppRunner : isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}   
2017-04-23 18:17:09.481 INFO 8054 — [ main] forezp.AppRunner : isbn-1234 –>Book{isbn=’isbn-1234’, title=’Some book’}

只有打印前面2个数据，程序等了3s，之后的数据瞬间打印在控制台上了，这说明缓存起了作用。

源码下载：<https://github.com/forezp/SpringBootLearning>

参考资料

[caching](https://spring.io/guides/gs/caching/)

[Spring Boot 揭秘与实战（二） 数据缓存篇 - 快速入门](http://blog.720ui.com/2017/springboot_02_data_cache_concurrenmapcache/)

优秀文章推荐：

* 更多springboot 教程：[springBoot非官方教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70341818)
* 更多springcoud 教程：[史上最简单的 SpringCloud 教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70148833)