

## 90\_代码实战：实现教师信息更新缓存过期逻辑

**儒猿架构官网上线**，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：[www.ruyuan2020.com](http://www.ruyuan2020.com)（建议 PC 端访问）

### 1、开篇

上节课首先分析了批量处理任务在更新教师信息的同时并没有更新教师列表缓存的情况，通过对旧有流程的分析发现，我们需要加入清除教师列表缓存的功能。这样在消费者下次请求教师列表的时候就会应为缓存为空，而主动从数据库获取教师的最新信息，并且将这些信息放入到缓存中。本节课是代码实战的环节，我们将用代码实现教师信息更新缓存过期逻辑。今天课程的内容包括以下几个部分：

- 回顾查询教师列表缓存功能
- 执行任务时清除教师列表缓存

### 2、回顾查询教师列表缓存功能

如图 1 所示，在 TeacherServiceImpl 中 getPage 方法上面的 Cacheable 注释，其中加入 condition 并且用 SpEL 对 teacher 进行逻辑判断。

teacher.page<=2 && teacher.teacherName.empty ,这句话的意思就是当教师列表的页数小于等于 2 页的时候，并且搜索教师姓名为空的时候，对教师列表进行缓存。

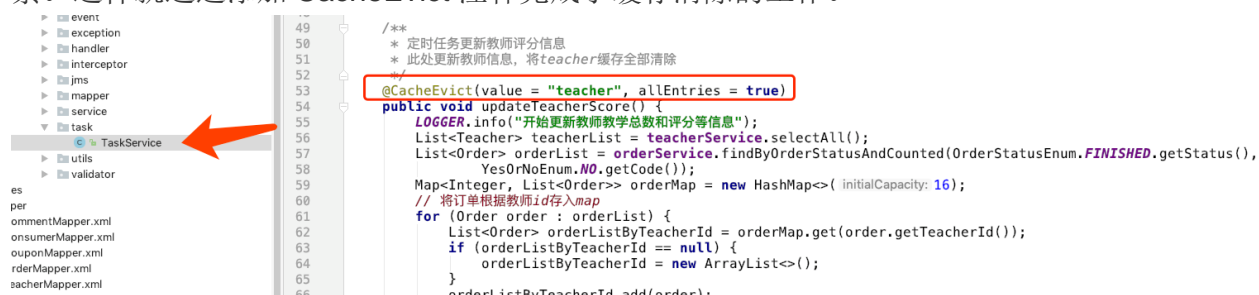


图 1 TeacherServiceImpl 对教师列表的缓存

通过对教师列表缓存代码的回顾，我们发现只要方法执行到 `getPage`，并且满足 SpEL 定义的条件就会获取缓存中的数据，除非缓存里面没有保存任何数据。实际上我们需要做的就是更新教师信息的时候将缓存数据清空，在此处调用的时候就不会走缓存的逻辑，而是通过 `teacherMapper` 中的 `getPage` 方法从数据库中读取教师信息了。

### 3、执行任务时清除教师列表缓存

回顾了教师列表的缓存代码之后，将镜头摇回 `TaskService` 中，上节课中介绍了这个 `Task` 是用来定时更新教师得分和教授课程的计划任务，每 5 分钟会运行一次。如图 2 所示，我们在 `updateTeacherScore` 方法上面打上了 `@CacheEvict` 的注释，这个注释的意思是清除缓存。并且指定 `value` 为“`teacher`”，说明需要清除“`teacher`”中的缓存，同时还定义了 `allEntries = true`，表示清楚缓存中所有的元素。这样就通过添加 `CacheEvict` 注释完成了缓存清除的工作。

图 2 在 `updateTeacherScore` 方法上清除缓存

这里需要特别说明的是，`@CacheEvict` 是用来清除缓存的。实际上它就是一个触发器，每次调用被它注解的方法时，就会触发清除指定缓存的动作。跟 `@Cacheable` 和 `@CachePut` 一样，`@CacheEvict` 也要求指定一个或多个缓存，也指定自定义一的缓存解析器和 `key` 生成器，也支持指定条件（`condition` 参数）。

此处我们使用了 `allEntries` 作为清除缓存的条件，它是 `Boolean` 类型，表示是否需要清除缓存中的所有元素。默认为 `false`，表示不需要。当指定了 `allEntries` 为 `true` 时，Spring Cache 将忽略指定的 `key`。有的时候我们需要 `Cache` 一下清除所有的元素，这比一个一个清除元素更有效率。

#### 4、总结

本节课程先回顾了教师列表的缓存代码，发现只要满足缓存条件，并且缓存中有数据就会从缓存中获取对应的教师信息。因此，需要在 `TaskService` 中的 `updateTeacherScore` 方法上打上 `EvictCache` 的注释，说明要清除的缓存以及指定清除缓存中所有元素。这样当计划任务定时执行的时候，就会因为 `EvictCache` 注释的加持而自动清除缓存的内容。当查询教师列表时，发现缓存内容为空就会从数据库中获取教师的最新数据。

下节课是将执行代码，更新缓存过期逻辑验证任务调度执行后，教师信息是否更新、缓存是否过期。这里将[代码](#)给大家，下期见，拜拜。

友情提示：本章讲述代码只是部分核心代码，完整代码请查阅文末链接中代码，谢谢。