

## 58\_代码实战：基于 Spring Cache 优化首页教师列表数据的热点查询功能

---

**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

上节课对高并发下首页教师列表的数据热点查询问题进行了分析，并且提出了解决的思路。对于不经常修改，有被经常查询的教师列表数据可以采取缓存的方式进行保存。于是，修改了之前定义的业务流程，加入了缓存的部分，通过 Spring Cache 保存教师列表，在下次访问该信息的时候就可以从缓存中读取，提高系统性能。本节课会聊聊，如何解决高并发下首页教师列表的数据热点查询问题。并且通过代码实现 Spring Cache 优化首页教师列表数据的热点查询功能。今天课程的内容包括以下几个部分：

- Teacher 实体类定义
- TeacherMapper 定义
- TeacherService 和 TeacherServiceImpl 定义
- TeacherKeyGenerator 定义
- TeacherController 定义

### 2、Teacher 实体类定义

由于教师列表查询需要用到 Teacher 作为查询展示的实体，因此这里需要定义 Teacher 实体类。如图 1 所示，在/spring/dto 目录下面定义 Teacher 实体类，这个类主要是给返回的教师列表使用。因此有 startIndex 这个是列表开始的索引，endIndex 是返回列表的长度。随后，还有 page 记录页数，size 记录每页包含多少条记录。再往后就是教师的基本信息，例如：Id、教师名称、学科、评分、教学天数等信息。

### 3、TeacherMapper 定义

如图 2 所示，在 `/spring/mapper` 目录下面定义 `TeacherMapper interface`，其中定义了 `findById`、`selectOne`、`findByName` 之类的通过 `id` 或者名字获取教师信息的方法。我们主要看 `getPage` 方法，主要传入 `teacher` 实体对象、以及 `orderColumn` 排序的列和 `orderType` 排序的类型（生序/降序）。

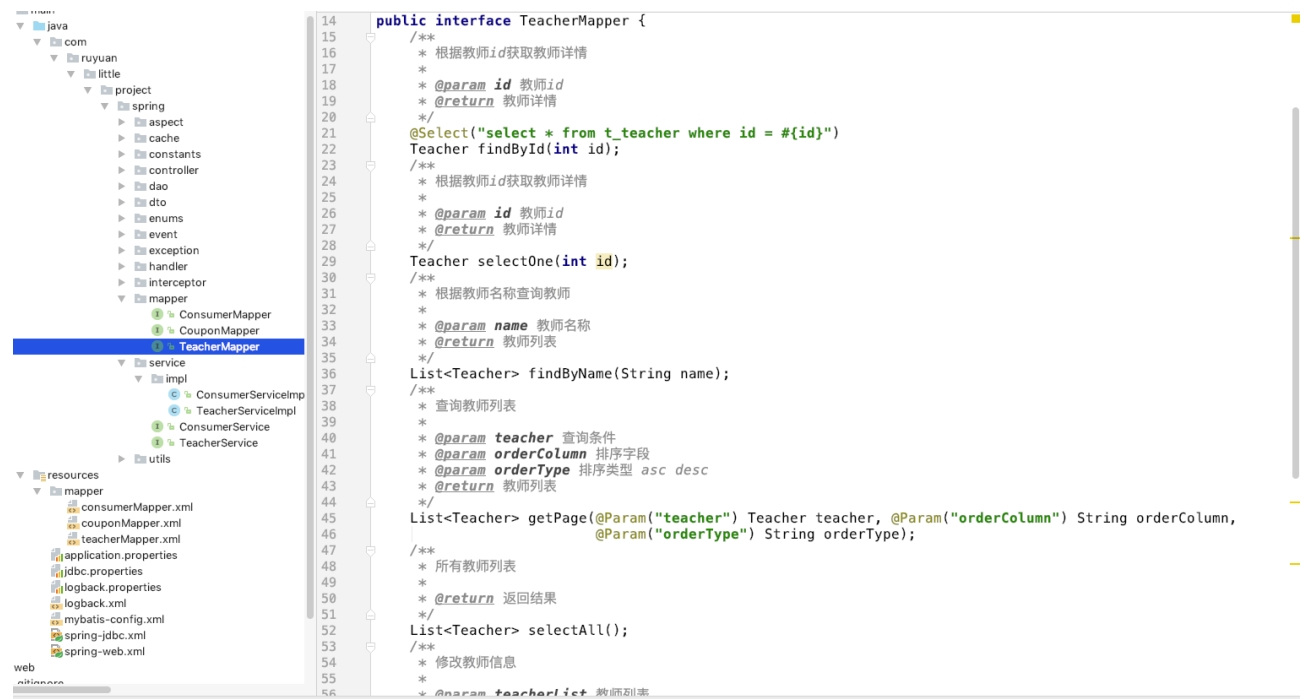


图 2 TeacherMapper 接口定义

看完 `TeacherMapper` 接口的定义，再来看对应的 `Mapper` 文件是如何实现接口的。

如图 3 所示，在 `/resources/mapper` 目录下面定义了 `teacherMapper.xml` 文件，该文件是对 `TeacherMapper` 接口的具体实现。也是通过配置 SQL 语句对数据库进行访问，这里来看 `getPage` 方法它通过对 `t_teacher` 的 `select` 查询返回需要的 `teacher` 信息。其中会针对传入的 `orderColumn` 和 `orderType` 字段进行 `order by` 操作。也会根据 `startIndex` 和 `EndIndex` 对搜索出来的结果进行分页返回。

#### 4、TeacherService 和 TeacherServiceImpl 定义

有 **Teacher** 实体和访问数据库的接口以后，再来看在服务层如何调用它们。如图 4 所示，在 `spring/service` 目录下面创建 **TeacherService** 的访问接口，然后在 `spring/service/impl` 目录下面创建对应的实现 **TeacherServiceImpl**，从该实现中可以看出主要是调用 **TeacherMapper** 中的方法直接访问数据库。对于我们需要缓存的 `getPage` 方法，从图上可以看出使用了 `@Cacheable` 的 annotation，`value` 用的是 `teacher`，`keyGenerator` 使用的 `teacherKeyGenerator`。方法体内部主要是通过 `teacherMapper` 调用 `getPage` 方法返回教师列表，如果列表不为空就对其进行遍历并且返回给调用者。由于 `@Cacheable` 的加入对列表的内容进行了缓存。

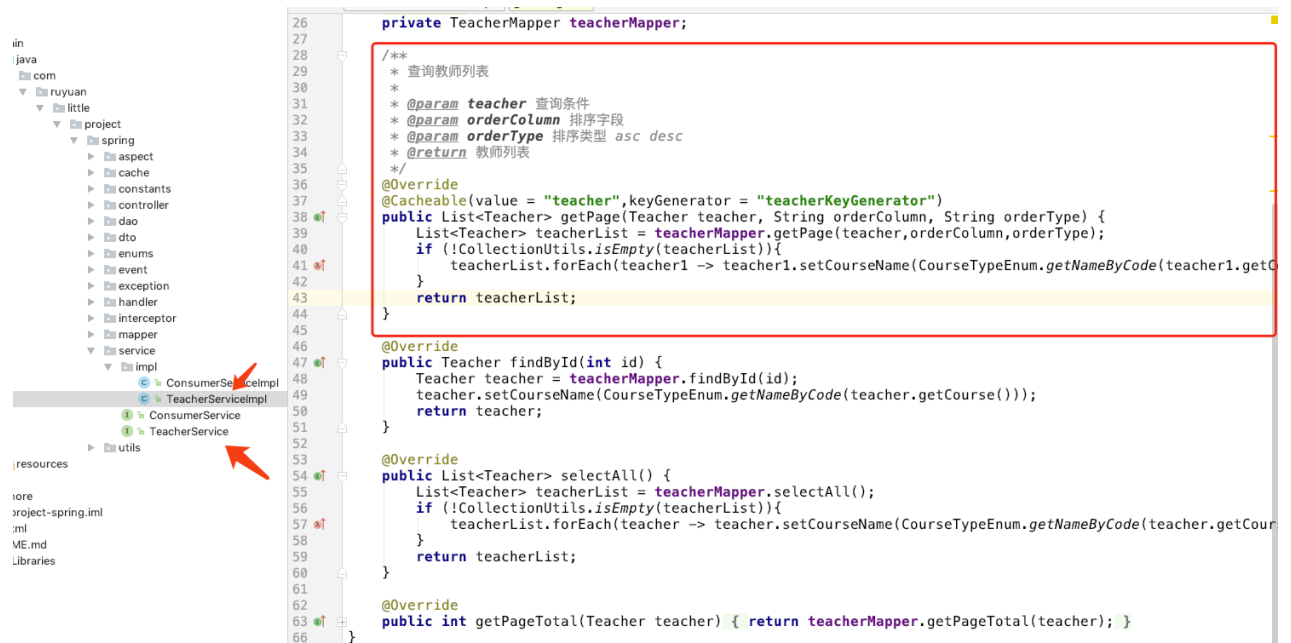


图 4 TeacherServiceImpl 实现和 TeacherService 接口

## 5. TeacherKeyGenerator 定义

上一部分提到的 teacherKeyGenerator 是一个自定义的 KeyGenerator，从字面上的意思可以理解为 Key 的生成器。由于缓存的数据多种多样，Spring Cache 在缓存的时候为了对缓存进行区别就要依靠这个 Key。这里的 KeyGenerator 它负责为缓存中的每个数据项生成每个键，这些键将用于在检索时查找数据项。

如图 5 所示，在 spring/cache 目录下面建立 TeacherKeyGenerator 类，该类实现了 KeyGenerator，并且 override 了 generate 方法。传入 Object、Method 和 Object 的数组作为输入参数。方法体内主要是生成 Key 的规则，并且返回给调用者。

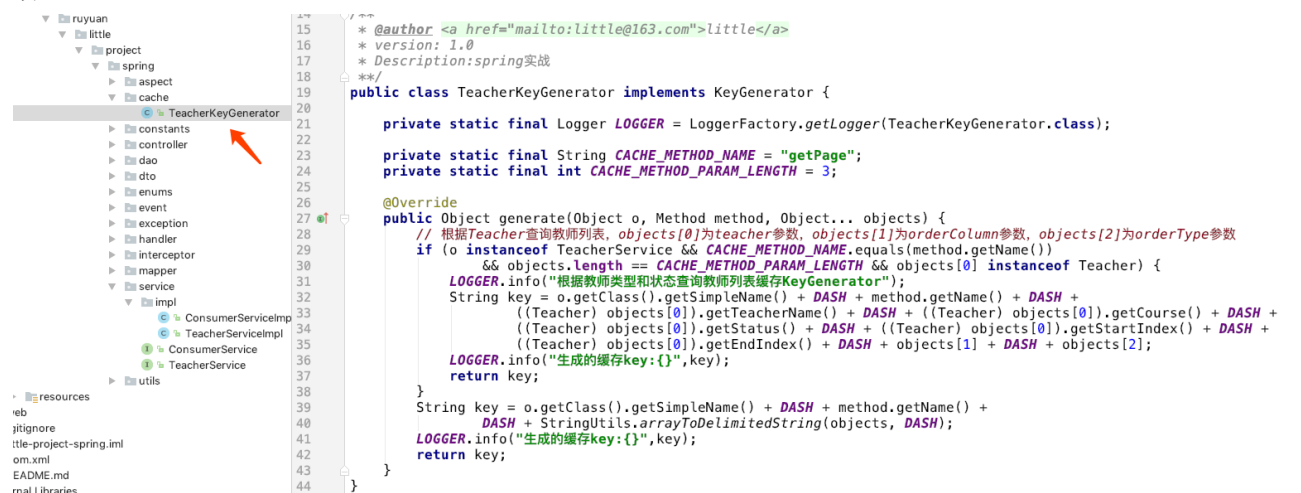


图 5 TeacherKeyGenerator 定义

注意生成 Key 的规则可以根据具体的业务来定义，图中生成 Key 的方法是，通过类名+方法名+课程名字+教师状态+页面开始索引+页面获取长度组成的。

## 6、TeacherController 定义

定义实体、数据库访问、服务以及缓存的类之后，来到调用接口的部分了。如图 6 所示，在 spring/controller 目录下创建 TeacherController 类，其中定义了 getPage 方法，路径也是 getPage。方法中主要是调用 TeacherService 中的 getPageToal 方法传入对应的参数，并将结果填充到儒猿封装的 TableData 类中返回给前端程序。

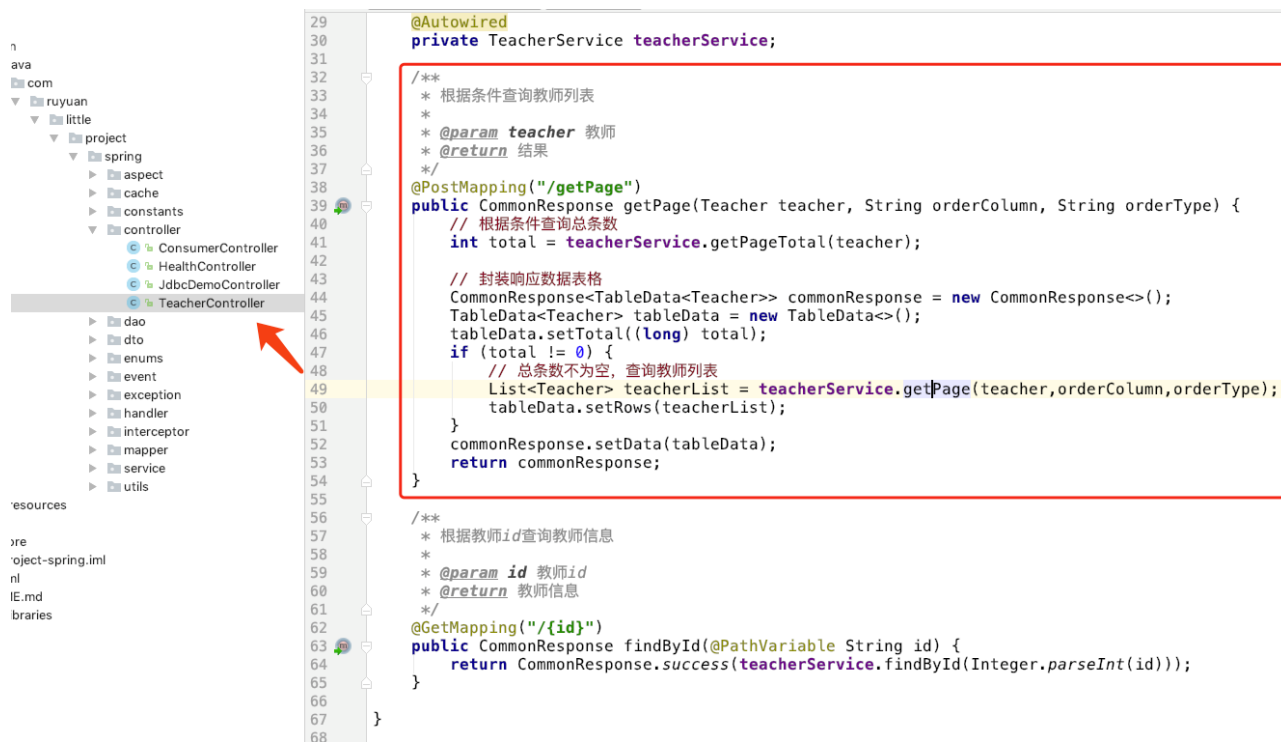


图 6 TeacherController 定义

## 7、总结

本节课通过介绍教师列表的代码实现，把 Teacher 的实体类、TeacherMapper 的数据库访问、TeacherService 的服务、TeacherKeyGenerator 的缓存 Key 生成规则以及 TeacherController 的访问接口，逐一给大家做了介绍。下节课我们会对代码进行测试，看看运行结果如何。这里将[代码](#)给大家，下期见，拜拜。