

## 61\_代码实战：基于 Spring SpEL 机制，只缓存前十条教师记录

---

**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

上节课思考当缓存数据过多情况下系统负荷会增大，然后通过分析教师列表的应用场景，发现不用缓存所有的教师信息，只需要按照一定规则缓存部分信息就可以满足消费者的需求。这个规则是值缓存前面 2 页的教师信息，只缓存教师列表不缓存教师搜索的信息。这节课我们会带大家修改缓存的代码，看看如何使用 Spring SpEL 机制缓存前十条教师记录。今天课程的内容包括以下几个部分：

- Spring SpEL 介绍
- 修改 TeacherServiceImpl

### 2、Spring SpEL 介绍

在对 TeacherServiceImpl 改造之前先介绍一下 SpEL 的知识，因为在缓存判断条件的时候会用到它。

SpEL 是 Spring Expression Language 的简称，它是一种功能强大的表达式语言、用于在运行时查询和操作对象图；语法上类似于 Unified EL，但提供了更多的特性，特别是方法调用和基本字符串模板函数。

SpEL 支持字符表达式，布尔和关系操作符，正则表达式，类表达式，方法调用，关系操作符，赋值等功能。我们主要使用在 annotation 进行条件判断的功能，先上个例子让大家感受一下。

如图 1 所示，通过 Value 的 annotation 定义 SpEL，其内容是“#”开头用“{}”包裹的表达式。表达式的内容是 `systemProperties['file.encoding']`，用来获取系统文件的编码格式，然后将其值赋给 `encoding` 变量。如法炮制通过 `systemEnvironment['JAVA_HOME']` 将 Java 的目录赋值给 `home` 变量。

```

@Value("#{systemProperties['file.encoding']}")
private String encoding;
@Value("#{systemEnvironment['JAVA_HOME']}")
private String home;

```

图 1 通过 SpEL 获取系统参数和环境参数

再来看一个应用场景

如图 2 所示，在 `PreAuthorize` 的 annotation 中配置 SpEL。`PreAuthorize` 是用来判断权限的注释，从 SpEL 可以看出需要满足两个条件中的一个就可以访问 `findUsers` 方法。第一个是 `hasAuthority('ROLE_ADMIN')`，也就是访问者需要有 `ROLE_ADMIN` 的权限；第二个是访问者的名字和用户的名字相等。两个条件和起来就是，管理员可以查看任意用户信息，或者用户自己也可以查看自己的信息。

```

@PreAuthorize("hasAuthority('ROLE_ADMIN') or #reqVo.sysUser.username == #userDetails.username")
@PostMapping("/findUsers")
public Result<List<SysUser>> findUsers(@RequestBody FindUsersReqVo reqVo, @AuthenticationPrincipal UserDetails userDetails) {
    PageInfo<SysUser> pageInfo = userService.findUsers(reqVo);
    return new Result<>(pageInfo.getList(), pageInfo.getTotal());
}

```

图 2 `PreAuthorize` 应用 SpEL

### 3、修改 `TeacherServiceImpl`

实际上 SpEL 的应用范围很广，在这个项目中我们主要应用它在 annotation 中的逻辑判断功能。对我们上一讲提到的缓存的页数以及是否输入教师姓名进行逻辑限制。

如图 3 所示，在 `TeacherServiceImpl` 中修改 `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3 修改 TeacherServiceImpl

上面仅仅加入了缓存条件，没有对缓存生成 Key 和缓存数据本身的逻辑进行修改。如果在工业级别的应用场景，大家需要根据具体的业务场景对 condition 进行定义，这里给出例子让大家能够快速理解限制缓存的思路。

#### 4、总结

本节课主要介绍在代码级别如何对缓存数量进行限制，首先对 SpEL 做了简单介绍，着重介绍了在 Annotation 中的 SpEL 应用。随后根据缓存前 2 页的逻辑和搜索教师名字为空的条件对 Cacheable 注释进行了 SpEL 判断，从而达到限制缓存的目的。下节课我们会带大家把代码整体走一遍，测试一下缓存结果。这里将[代码](#)给大家，下期见，拜拜。