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 (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。

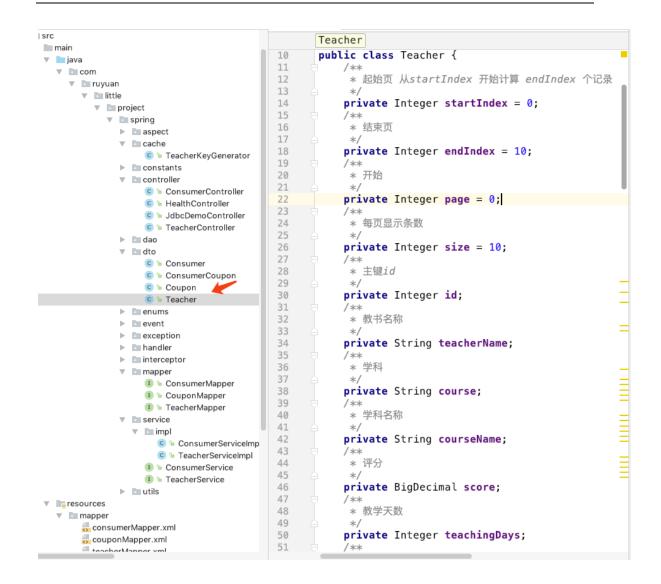


图 1 Teacher 实体类定义

3、TeacherMapper 定义

有了 Teacher 实体类以后需要从数据库中获取对应的实体并且填充到 Teacher 中去。由于我们使用 Mybatis 获取数据库信息,需要定义 TeacherMapper 的接口和对应的 XML 文件。

如图 2 所示,在/spring/mapper 目录下面定义 TeacherMapper interface,其中定义了 findByld、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)。

```
▼ injava
▼ incom
▼ intropusan
▼ intropiect
▼ intropiect
▼ intropiect
▼ intropiect
▼ intropiect
                                                          public interface TeacherMapper {
                                                 /**
* 根据教师id获取教师详情
                                                                 * @param id 教师id
                                                                 * @return 教师详情

▼ Image spring

                 aspect
                                                               @Select("select * from t_teacher where id = #{id}")
Teacher findById(int id);
                 ▶ □ cache
                 controller
dao
dto
                                                              /**
* 根据教师id获取教师详情
                                                                 * @param id 教师id
                   enums
                   event
                   m exception
                                                               Teacher selectOne(int id);
                   nandler handler
                                                                ** 根据教师名称查询教师
                                                                * @param name 教师名称
                                                                 * @return 教师列表
                   service
                                                               List<Teacher> findByName(String name);
                    ▼ 🛅 impl
                      © to ConsumerServiceIm
© to TeacherServiceImpl

to ConsumerService

to TeacherService
                                                                * 查询教师列表
                                                                * @param teacher 查询条件
* @param orderColumn 排序字段
* @param orderType 排序类型 asc desc
* @return 教师列表
                 ▶ b utils
 ▼ lieresources
     mapper
consumerMapper.xml
cocouponMapper.xml
teacherMapper.xml
                                                              application.properties
                                                                 * 所有教师列表
       idbc.properties
       logback.properties
                                                                  · @return 返回结果
      mybatis-config.xml
spring-jdbc.xml
spring-web.xml
                                                               List<Teacher> selectAll();
                                                                * 修改教师信息
                                                                 * Anaram toacharlist 教師別書
```

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 orderColum 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。

```
src

main

m
                                                                                                                                                                                                                                                                        <select id="getPage"
    resultType="com.ruyuan.little.project.spring.dto.Teacher">
    select * from t_teacher
                                                                                                                                                                                                                                                                                            where 1 = 1
<if test="teacher.id != null and teacher.id !=''">
                                                                                                                                                                                                                                                                                                                And id = \#\{id\}
                                                                                                                                                                                                                                                                                                cif test="teacher.teacherName != null and teacher.teacherName !=
And teacher_name like concat('%',#{teacher.teacherName},'%')
s/if>
                                                                                                                                                                                                                                                                                           <if test="teacher.course != null and teacher.course !=''"
And course = #{teacher.course}
</if>
                                                                                                                                                                                                                                                                                          </if>
</if test="teacher.score != null and teacher.score !='' and teacher.score >0">
And score = #{teacher.score}
</if>
</if test="teacher.teachingDays != null and teacher.teachingDays !='' and teacher.teachingDays >0">
And teaching_days = #{teacher.teachingDays}
</if>
</if test="teacher.complex">
</if te
                                                                                        🛅 dto
                                                                                        enums
                                                                                        event
                                                                                         exception
                                                                                        handler
interceptor
mapper
                                                                                                                                                                                                                                                                                           <if test="teacher.status != null and teacher.status !=''">
                                                                                                                                                                                                                                                                                             And status = #{teacher.status} </if>
                                                                                  ▶ □ utils
                                                                                                                                                                                                                                                                                                 //II>
if test="teacher.teachingCount != null and teacher.teachingCount !='' and teacher.teachingCount >0
And teaching_count = #{teacher.teachingCount}
//if>

    ▼ Impresources

And price_of_day = #{teacher.priceOfDay}

</
                                   mapper
                                                  aconsumerMapper.xml
                                        couponMapper.xml
teacherMapper.xml
                                                                                                                                                                                                                                                                                           <if test="teacher.description != null and teacher.description !=''">
And description = #{teacher.description}
</if>
                                         jdbc.properties
                                        logback.properties
                                        alogback.xml

'if test="teacher.detail != null and teacher.detail !=''">
And detail = #{teacher.detail}
</if>
                                         mybatis-config.xml

<

<if test="orderType != null and orderType !='">
</if>
</if>
</if>
</if>
                 .gitignore
            little-project-spring.iml
           README.md
            xternal Libraries
                                                                                                                                                                                                                                                                                                            iit #{teacher.startIndex},#{teacher.endIndex}
```

图 3 teacherMapper.xml d 定义

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。

```
26
27
28
29
30
31
32
33
34
35
36
37
38
40
41
42
43
44
45
                                                                                                                                                                                                                                                                                                                                private TeacherMapper teacherMapper;
in
java
com
ruyuan
little
litt
                                                                                                                                                                                                                                                                                                                                         * 查询教师列表
                                                                                                                                                                                                                                                                                                                                                   <u>@param</u> teacher 查询条件
<u>@param</u> orderColumn 排序字段
<u>@param</u> orderType 排序类型 asc desc
                                                                                                                                                                                                                                                                                                                                                   @return 教师列表
                                                                       aspectcache
                                                                           ▶ ■ constants
                                                                                                                                                                                                                                                                                                                             @Override
@Cacheable(value = "teacher",keyGenerator = "teacherKeyGenerator")
public List<Teacher> getPage(Teacher teacher, String orderColumn, String orderType) {
    List<Teacher> teacherList = teacherMapper.getPage(teacher,orderColumn,orderType);
    if (!CollectionUtils.isEmpty(teacherList)){
        teacherList.forEach(teacher1 -> teacher1.setCourseName(CourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getCourseTypeEnum.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.getNameByCode(teacher1.g

    □ controller

                                                                                        enums
                                                                                      event
                                                                                      exception
                                                                                      nandle
                                                                                      interceptor
mapper
                                                                                        service
                                                                                                                                                                                                                                                                                                                                public Teacher findById(int id) {
   Teacher teacher = teacherMapper.findById(id);
   teacher.setCourseName(CourseTypeEnum.getNameByCode(teacher.getCourse()));
   return teacher;
                                                                                                                                                                                                                                                       47 af
                                                                                                                    © 1 ConsumerSe celmpl
                                                                                               © % TeacherServiceImpl

    ConsumerService
    TeacherService
    utils
                                                                                                                                                                                                                                                  51
52
53
54 ®1
55
56
57 &1
58
                                                                                                                                                                                                                                                                                                                               public List<Teacher> selectAll() {
   List<Teacher> teacherList = teacherMapper.selectAll();
   if (!CollectionUtils.isEmpty(teacherList)){
             project-spring.iml
ml
ME.md
                                                                                                                                                                                                                                                                                                                                                                                       teacherList.forEach(teacher -> teacher.setCourseName(CourseTypeEnum.getNameByCode(teacher.getCour
                                                                                                                                                                                                                                                  58
59
60
61
62
63 ©
                                                                                                                                                                                                                                                                                                                                                             return teacherList:
                                                                                                                                                                                                                                                                                                                               @Override
public int getPageTotal(Teacher teacher) { return teacherMapper.getPageTotal(teacher); }
```

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。
    ▼ Paruyuan
▼ Ittle
▼ Paruyuan
▼ Spring
                                                * <u>@author</u> <a href="mailto:little@163.com">little</a>
* version: 1.0
               * Description:spring实战
            ▶ □ aspect▼ □ cache
                                                public class TeacherKeyGenerator implements KeyGenerator {
                                                    private static final Logger LOGGER = LoggerFactory.getLogger(TeacherKeyGenerator.class);
                                                    private static final String CACHE_METHOD_NAME = "getPage";
private static final int CACHE_METHOD_PARAM_LENGTH = 3;
               enums
event
                                       25
26
27 1
28
29
30
31
                                                    exception
               la handler
               interceptor
               mapper
service
                                                              return key;
             ▶ 🖿 utils
 resources
                                                         String key = o.getClass().getSimpleName() + DASH + method.getName() + DASH + StringUtils.arrayToDelimitedString(objects, DASH);
LOGGER.info("生成的缓存key:{}",key);
                                                         return key;
```

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。

```
29
30
31
32
33
34
35
36
37
38
39
                                                                private TeacherService teacherService;
ava
                                                                    根据条件查询教师列表
▼ 🛅 ruyuan
                                                                    <u>@param</u> teacher 教师
<u>@return</u> 结果

    ▼ Improject

           spring
            aspect
                                                                @PostMapping("/getPage")
            ▶ □ cache
                                                               public CommonResponse getPage(Teacher teacher, String orderColumn, String orderType) {
// 根据条件查询总条数
               constants
              controller
                                                                      int total = teacherService.getPageTotal(teacher);
                  © 1 ConsumerController
                                                42
43

    □ HealthController

                                                                      // 封装响应数据表格
                   © 🚡 JdbcDemoControlle
                                                                      // 封表明版及版本符
CommonResponse<TableData<Teacher>> commonResponse = new CommonResponse<>();
TableData<Teacher> tableData = new TableData<>();
tableData.setTotal((long) total);
if (total != 0) {
    // 总条数不为空,查询教师列表
    List<Teacher> teacherList = teacherService.getPage(teacher,orderColumn,orderType);
    tableData.setRows(teacherList):
                                                44
45
                  C TeacherController
                                                46
            b dto
              enums
                                                48
               event
                                               49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
              exception
                                                                            tableData.setRows(teacherList):
               handler
            ▶ interceptor
                                                                      commonResponse.setData(tableData);
               mapper mapper
                                                                      return commonResponse;
               service
'esources
                                                                  * 根据教师id查询教师信息
oject-spring.iml
                                                                    @param id 教师id
                                                                    @return 教师信息
IE.md
braries
                                                               @GetMapping("/{id}")
public CommonResponse findById(@PathVariable String id) {
                                                                      return CommonResponse.success(teacherService.findById(Integer.parseInt(id)));
                                               66
67
68
```

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。