定义了一个多对多关系生成表的getbyid方法和updatebyid方法

@Select("SELECT \* FROM teacher\_course WHERE cid = #{cid} AND tid = #{tid}")  
Teacher\_Course getRelation(@Param("cid") Integer cid, @Param("tid") Integer tid);  
  
@Update("UPDATE teacher\_course SET cnt = #{cnt} WHERE cid = #{cid} AND tid = #{tid}")  
Integer updateCntById(@Param("cid") Integer cid, @Param("tid") Integer tid, @Param("cnt") Integer cnt);

并在service层实现了

资讯界面给前端提供宿舍和院校信息

package cn.edu.cqu.nsers.Controller;  
  
import cn.edu.cqu.nsers.Service.DepartmentService;  
import cn.edu.cqu.nsers.Service.DormitoryService;  
import cn.edu.cqu.nsers.pojo.Dormitory;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
  
import java.util.HashMap;  
import java.util.Map;  
  
@Controller  
@RequestMapping("")  
public class News {  
 @Autowired  
 public DepartmentService departmentService;  
 @Autowired  
 public DormitoryService dormitoryService;  
  
 @GetMapping("/newsPage")  
 public String showNewsPage() {  
 return "newsPage";  
 }  
  
 @PostMapping("/getInfo")  
 public ResponseEntity<Map<String,Object>> getInfo()  
 {  
 Map<String, Object> response = new HashMap<>();  
 int cnt1 = departmentService.getById(1).getDepartmentCnt();  
 int cnt2 = departmentService.getById(2).getDepartmentCnt();  
 int cnt3 = departmentService.getById(3).getDepartmentCnt();  
 long total1 = dormitoryService.getSumOfGarden("梅园");  
 long total2 = dormitoryService.getSumOfGarden("竹园");  
 response.put("success",true);  
 response.put("Computer",cnt1);  
 response.put("Mathematics",cnt2);  
 response.put("Philosophy",cnt3);  
 response.put("Plumgarden",total1);  
 response.put("Bamboogarden",total2);  
 return ResponseEntity.*ok*(response);  
 }  
  
}

更新了选课的功能，更加严谨，确保了老师的唯一性

package cn.edu.cqu.nsers.Controller;  
  
import cn.edu.cqu.nsers.Mapper.TeacherMapper;  
import cn.edu.cqu.nsers.Service.CourseService;  
import cn.edu.cqu.nsers.Service.StudentService;  
import cn.edu.cqu.nsers.Service.TeacherService;  
import cn.edu.cqu.nsers.Service.Teacher\_CourseService;  
import cn.edu.cqu.nsers.pojo.Course;  
import cn.edu.cqu.nsers.pojo.Teacher;  
import cn.edu.cqu.nsers.pojo.Teacher\_Course;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.stereotype.Repository;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.concurrent.Callable;  
  
@Controller  
@RequestMapping("/function/course")  
public class Func\_Course {  
 @Autowired  
 public StudentService studentService;  
 @Autowired  
 public CourseService courseService;  
 @Autowired  
 public TeacherService teacherService;  
 @Autowired  
 public Teacher\_CourseService teacher\_courseService;  
 @Autowired  
 public TeacherMapper teacherMapper;  
  
 @RequestMapping("/getCourse")  
 public String showCourse(Model model) //在查课页面一开始展示所有课程信息+授课教师  
 {  
 List<Map<String, Object>> course = courseService.listMaps();  
 List<Map<String, Object>> teacher = teacherService.listMaps();  
 model.addAttribute("course",course);  
 model.addAttribute("teacher",teacher);  
 return "getCoursePage";  
 }  
  
 @PostMapping("/getCourse/{cname}")  
 public String getCourse(@PathVariable("cname") String cname ,Model model) //查课时根据学生搜索的课程名称返回课程信息+授课教师  
 {  
 QueryWrapper<Course> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("cname", cname);  
 Course course = courseService.getOne(queryWrapper);  
 int cid = course.getCid();  
 QueryWrapper<Teacher> teacherQueryWrapper = new QueryWrapper<>();  
 teacherQueryWrapper.eq("cid",cid);  
 List<Teacher> teacherList = teacherMapper.selectList(teacherQueryWrapper);  
 model.addAttribute("course",course);  
 model.addAttribute("teacherList",teacherList);  
 return "getCoursePage";  
 }  
  
 @PostMapping("/showSelectCourse")  
 public String selectCourse(Model model) { //在选课页面一开始展示所有课  
 List<Course> courseList = courseService.list();  
 model.addAttribute(courseList);  
 return "selectCoursePage";  
 }  
  
 @PostMapping("/showSelectCourse/{cname}")  
 @ResponseBody  
 public List<Teacher\_Course> select(@PathVariable("cname") String cname,  
 @RequestParam(value = "tname", required = false) String tname,  
 Model model)  
 //选课操作  
 {  
 if (tname != null) {  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname", cname);  
 Course course = courseService.getOne(courseQueryWrapper);  
 course.setCcnt(course.getCcnt() + 1);  
 courseService.updateById(course);  
 int cid = course.getCid();  
 QueryWrapper<Teacher> teacherQueryWrapper = new QueryWrapper<>();  
 teacherQueryWrapper.eq("tname", tname);  
 Teacher teacher = teacherService.getOne(teacherQueryWrapper);  
 int tid = teacher.getTid();  
 Teacher\_Course teacher\_course = teacher\_courseService.getRelation(cid, tid);  
 int total = teacher\_courseService.updateCntById(cid,tid,teacher\_course.getCnt()+1);  
 List<Teacher\_Course> l = teacher\_courseService.list();  
 return l;  
 }  
 else  
 {  
 QueryWrapper<Course> courseQueryWrapper = new QueryWrapper<>();  
 courseQueryWrapper.eq("cname",cname);  
 Course course = courseService.getOne(courseQueryWrapper);  
 QueryWrapper<Teacher\_Course> teacher\_courseQueryWrapper = new QueryWrapper<>();  
 teacher\_courseQueryWrapper.eq("cid", course.getCid());  
 List<Teacher\_Course> teacher\_courseList = teacher\_courseService.list(teacher\_courseQueryWrapper);  
 model.addAttribute("teacher\_courseList", teacher\_courseList);  
 return teacher\_courseList;  
 }  
 }  
  
 @RequestMapping("/showSelectCourse/searchCourse/{cname}")  
 public String Search(@PathVariable("cname") String cname, Model model) { //选课时根据学生搜索的课程名称返回课程信息+授课教师  
  
 QueryWrapper<Course> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("cname", "cname");  
 Course course = courseService.getOne(queryWrapper);  
 Teacher\_Course teacher\_course = teacher\_courseService.getById(course.getCid());  
 Teacher teacher = teacherService.getById(teacher\_course.getTid());  
 model.addAttribute("course",course);  
 model.addAttribute("teacher",teacher);  
 return "selectCoursePage";  
 }  
}