前台页面展示

1. 首页

**轮播图（展示最热的三门课程和最新的三门课程），参考代码**：

public List<CourseVo> recommend() {  
 List<CourseVo> courseVos = WhiteListCache.cache.getIfPresent("coursesRecommend") == null ? new ArrayList<>() : WhiteListCache.cache.getIfPresent("coursesRecommend");  
 if (CollectionUtils.isNotEmpty(courseVos)) {  
 return courseVos;  
 }  
 //获取课程学习最热前三条记录  
 StringBuffer sql = new StringBuffer("SELECT course\_id,COUNT(student\_id) student from s\_course\_student GROUP BY course\_id ORDER BY student desc");  
 List<Object> params = new ArrayList<>();  
 sql.append(" limit ?,?");  
 params.add(1);  
 params.add(3);  
 List<CourseStudent> courseStudents = jdbcTemplate.query(sql.toString(), params.toArray(), new BeanPropertyRowMapper<>(CourseStudent.class));  
 if (CollectionUtils.isNotEmpty(courseStudents)) {  
 for (CourseStudent courseStudent : courseStudents) {  
 Course course = courseRepository.getOne(courseStudent.getCourseId());  
 if (course != null) {  
 CourseVo courseVo = new CourseVo();  
 courseVo.setId(course.getId());  
 courseVo.setCourseImageUrl(course.getCourseImageUrl());  
 courseVos.add(courseVo);  
 }  
 }  
 }  
 //获取最新的三条课程信息  
 StringBuffer newSql = new StringBuffer("select \* from s\_course order by create\_time desc");  
 newSql.append(" limit ?,?");  
 List<CourseVo> courseVoList = jdbcTemplate.query(newSql.toString(), params.toArray(), new BeanPropertyRowMapper<>(CourseVo.class));  
 courseVos.addAll(courseVoList);  
 WhiteListCache.cache.put("coursesRecommend", courseVos);  
 return courseVos;  
}

精品课程代码跟后台的课程列表代码是一样的

1. 作业列表

**学生提交作业功能（学生下载老师布置的作业，答题完后上传提交）**

public HomeworkSubmit submit(HomeworkSubmit homeworkSubmit) {  
 HomeworkSubmit submit = homeworkSubmitRepository.findByHomeworkIdAndUserId(homeworkSubmit.getHomeworkId(), GaeaContext.getUserId());  
 if (submit == null) {  
 submit = new HomeworkSubmit();  
 submit.setCreateTime(new Date());  
 }  
 submit.setHomeworkId(homeworkSubmit.getHomeworkId());  
 submit.setHomeworkFilePath(homeworkSubmit.getHomeworkFilePath());  
 submit.setUserId(GaeaContext.getUserId());  
 return homeworkSubmitRepository.save(submit);  
}

1. 课程详情

**开始学习功能**

public void study(Integer id, Integer chapterId) throws Exception {  
 UserStudyProgress userStudyProgress = userStudyProgressRepository.findByCourseIdAndUserIdAndChapterId(id, GaeaContext.getUserId(), chapterId);  
 if (userStudyProgress == null) {  
 userStudyProgress = new UserStudyProgress();  
 userStudyProgress.setCourseId(id);  
 userStudyProgress.setChapterId(chapterId);  
 userStudyProgress.setProgress(0);  
 userStudyProgress.setUserId(GaeaContext.getUserId());  
 userStudyProgress.setCreateTime(new Date());  
 } else {  
 userStudyProgress.setCreateTime(new Date());  
 }  
 userStudyProgressRepository.save(userStudyProgress);  
}

评价功能

发布评价功能代码

public CourseCommentVo create(CourseCommentVo courseCommentVo) {  
 CourseComment courseComment = new CourseComment();  
 BeanUtils.copyProperties(courseCommentVo, courseComment);  
 courseComment.setUserId(GaeaContext.getUserId());  
 courseComment.setCreateTime(new Date());  
 courseCommentRepository.save(courseComment);  
 return courseCommentVo;  
}

列表功能代码

public PagerHelper<CourseCommentVo> list(String key, Integer userId, Integer courseId, Integer pageNo, Integer pageSize) {  
 PagerHelper<CourseCommentVo> result = new PagerHelper<>();  
 StringBuffer sql = new StringBuffer("select \* from s\_course\_comment where 1=1");  
 List<Object> params = new ArrayList<>();  
 if (StringUtils.isNotEmpty(key)) {  
 sql.append(" and course\_name like ?");  
 params.add("%" + key + "%");  
 }  
 if (courseId != null) {  
 sql.append(" and course\_id = ?");  
 params.add(courseId);  
 }  
 if (userId != null) {  
 sql.append(" and user\_id = ?");  
 params.add(userId);  
 }  
 StringBuffer totalSql = new StringBuffer("select count(1) from (" + sql.toString() + ") a");  
 Integer total = jdbcTemplate.queryForObject(totalSql.toString(), params.toArray(), Integer.class);  
  
 int startIndex = (pageNo - 1) \* pageSize;  
 sql.append(" limit ?,?");  
 params.add(startIndex);  
 params.add(pageSize);  
 List<CourseCommentVo> courseVos = jdbcTemplate.query(sql.toString(), params.toArray(), new BeanPropertyRowMapper<>(CourseCommentVo.class));  
 if (CollectionUtils.isNotEmpty(courseVos)) {  
 Map<Integer, User> userMap = userService.getAllUserMap();  
 Map<Integer, Course> courseMap = getCourseMap();  
 for (CourseCommentVo courseVo : courseVos) {  
 courseVo.setUserName(courseVo.getUserId() == null || userMap.get(courseVo.getUserId()) == null ? "" : userMap.get(courseVo.getUserId()).getUserName());  
 courseVo.setPhotoPath(courseVo.getUserId() == null || userMap.get(courseVo.getUserId()) == null ? "" : userMap.get(courseVo.getUserId()).getPhotoPath());  
 courseVo.setCourseName(courseVo.getCourseId() == null || courseMap.get(courseVo.getCourseId()) == null ? "" : courseMap.get(courseVo.getCourseId()).getCourseName());  
 courseVo.setCoursePath(courseVo.getCourseId() == null || courseMap.get(courseVo.getCourseId()) == null ? "" : courseMap.get(courseVo.getCourseId()).getCourseImageUrl());  
 }  
 }  
 result.setTotal(total);  
 result.setItems(courseVos);  
 return result;  
}

提问功能

发布提问功能

public QuestionVo create(QuestionVo questionVo) {  
 Question question = new Question();  
 BeanUtils.copyProperties(questionVo, question);  
 question.setUserId(GaeaContext.getUserId());  
 question.setCreateTime(new Date());  
 questionRepository.save(question);  
 return questionVo;  
 }

提问列表

public PagerHelper<QuestionVo> list(String key, Integer userId, Integer courseId, Integer pageNo, Integer pageSize) {  
 PagerHelper<QuestionVo> result = new PagerHelper<>();  
 StringBuffer sql = new StringBuffer("select \* from s\_question where 1=1");  
 List<Object> params = new ArrayList<>();  
 if (StringUtils.isNotEmpty(key)) {  
 sql.append(" and question\_content like ?");  
 params.add("%" + key + "%");  
 }  
 if (courseId != null) {  
 sql.append(" and course\_id = ?");  
 params.add(courseId);  
 }  
 if (userId != null) {  
 sql.append(" and user\_id = ?");  
 params.add(userId);  
 }  
 StringBuffer totalSql = new StringBuffer("select count(1) from (" + sql.toString() + ") a");  
 Integer total = jdbcTemplate.queryForObject(totalSql.toString(), params.toArray(), Integer.class);  
  
 int startIndex = (pageNo - 1) \* pageSize;  
 sql.append(" limit ?,?");  
 params.add(startIndex);  
 params.add(pageSize);  
 List<QuestionVo> questionVos = jdbcTemplate.query(sql.toString(), params.toArray(), new BeanPropertyRowMapper<>(QuestionVo.class));  
 if (CollectionUtils.isNotEmpty(questionVos)) {  
 Map<Integer, User> userMap = userService.getAllUserMap();  
 Map<Integer, Course> courseMap = getCourseMap();  
 for (QuestionVo questionVo : questionVos) {  
 questionVo.setUserName(questionVo.getUserId() == null || userMap.get(questionVo.getUserId()) == null ? "" : userMap.get(questionVo.getUserId()).getUserName());  
 questionVo.setPhotoPath(questionVo.getUserId() == null || userMap.get(questionVo.getUserId()) == null ? "" : userMap.get(questionVo.getUserId()).getPhotoPath());  
 questionVo.setCourseName(questionVo.getCourseId() == null || courseMap.get(questionVo.getCourseId()) == null ? "" : courseMap.get(questionVo.getCourseId()).getCourseName());  
 questionVo.setCoursePath(questionVo.getCourseId() == null || courseMap.get(questionVo.getCourseId()) == null ? "" : courseMap.get(questionVo.getCourseId()).getCourseImageUrl());  
 questionVo.setAnswerVos(findAnswerByQuestionId(questionVo.getId(), userMap));  
 }  
 }  
 result.setTotal(total);  
 result.setItems(questionVos);  
 return result;  
}

1. 基本信息

修改密码参考代码

public User updatePwd(Integer id, UserPwdVo userPwdVo) throws Exception {  
 User tempUser = userRepository.getOne(id);  
 if (tempUser == null) {  
 throw new ServiceException("不存在【" + id + "】对应的用户");  
 }  
 if (!tempUser.getPassword().equals(userPwdVo.getOldPassword())) {  
 throw new ServiceException("原密码不对，请重新输入");  
 }  
 tempUser.setPassword(userPwdVo.getNewPassword());  
 return userRepository.save(tempUser);  
}

1. 我的学习

我的课程

public PagerHelper<CourseVo> findMyCourse(Integer pageNo, Integer pageSize) {  
 PagerHelper<CourseVo> result = new PagerHelper<>();  
 StringBuffer sql = new StringBuffer("select a.\* from s\_course a,s\_user\_study\_progress b where 1=1 and a.id = b.course\_id");  
 List<Object> params = new ArrayList<>();  
 sql.append(" and b.user\_id = ?");  
 params.add(GaeaContext.getUserId());  
 sql.append(" group by b.course\_id");  
  
 StringBuffer totalSql = new StringBuffer("select count(1) from (" + sql.toString() + ") a");  
 Integer total = jdbcTemplate.queryForObject(totalSql.toString(), params.toArray(), Integer.class);  
  
 int startIndex = (pageNo - 1) \* pageSize;  
 sql.append(" limit ?,?");  
 params.add(startIndex);  
 params.add(pageSize);  
 List<CourseVo> courseVos = jdbcTemplate.query(sql.toString(), params.toArray(), new BeanPropertyRowMapper<>(CourseVo.class));  
 result.setTotal(total);  
 result.setItems(courseVos);  
 return result;  
}