企业Java技术

**综合实验报告**

**题目：2-娱乐内容测评论坛**

|  |  |  |  |
| --- | --- | --- | --- |
| **小组号：2** | | | |
| **序号** | **班级** | **学号** | **姓名** |
| **1** | **信安1701** | **170120326** | **傅暮云** |

**计算机科学与技术学院**

1. **引言**

本项目主要是一个提供对于音乐，影视，游戏等艺术作品进行评测的一个论坛，具备一些综合性质的系统功能。

本质上是简单模仿实现诸如猫眼、豆瓣等评测平台的机制。

项目意义：

（1）通过该系统，为艺术作品提供一个不一样的评价区。

（2）提供对于选择娱乐产品的参考。

（3）其他扩展功能的实践。

**以下是项目设计说明，报告结尾有附录说明**

1. **需求分析**
   1. **主要说明**

本项目主要分为基础模块，用户管理

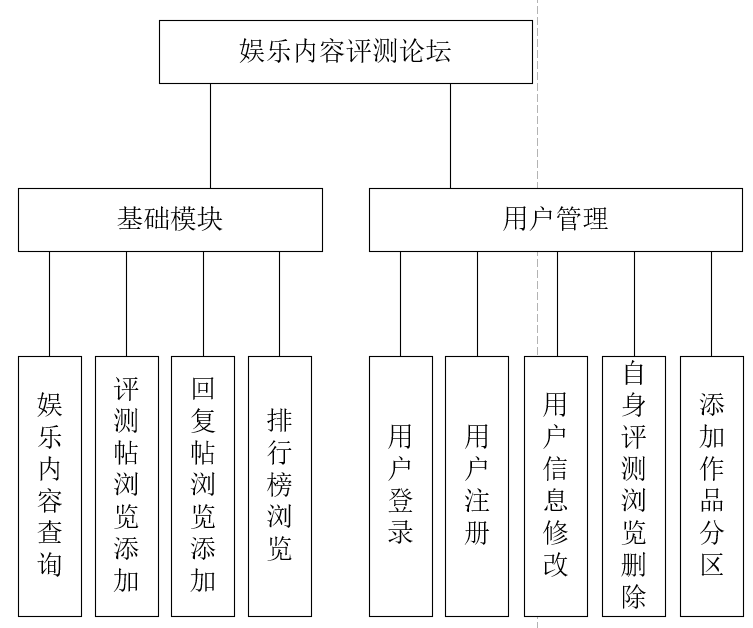
* 1. **功能需求**
     1. 基础模块

基础模块是本论坛的最基本功能，对于游客（即没有账号没有登录的用户）也可以访问使用其中一部分功能。功能如下：

1. 娱乐内容查询
2. 评测帖浏览和添加（添加需要登录）
3. 回复帖浏览和添加（添加需要登录）
4. 排行榜浏览
   * 1. 用户管理

用户管理是本论坛面向用户的功能区。对于登录了的用户可以在里面进行诸如用户信息的管理，以及一些对网站业务数据的增删改操作。功能如下：

1. 用户登录
2. 用户注册
3. 普通用户管理
4. 用户信息修改
5. 自身评测浏览和删除
6. 添加作品分区
   1. **层次结构图**



* 1. **技术选型&环境配置**

开发技术和环境说明：

WEB前端使用VSCode编辑器，技术选用Element UI框架（基于Vue框架）。同时使用了以下Vue插件辅助开发：

Vue-router：路由跳转

Vue-session：Session功能

Axios：Ajax技术的集成插件，实现异步更新和前后端数据交互

业务后端使用IDEA开发环境，技术选用SpringBoot框架。同时使用了以下组件进行开发：

Web：自带Tomcat服务器，SpringBoot支持Web开发的基本配置。

JPA：Java持久化API，通过这个可以面向对象地操作数据库。

MySQL：SpringBoot对于数据库选用的支持，可连接MySQL数据库。

运行环境：

服务器端MySQL 8.0.19，访问端WEB浏览器访问

1. **项目设计**
   1. **数据库设计**
      1. **概要设计**

实体：

主要模块：作品、评测帖、回复帖

用户模块：用户

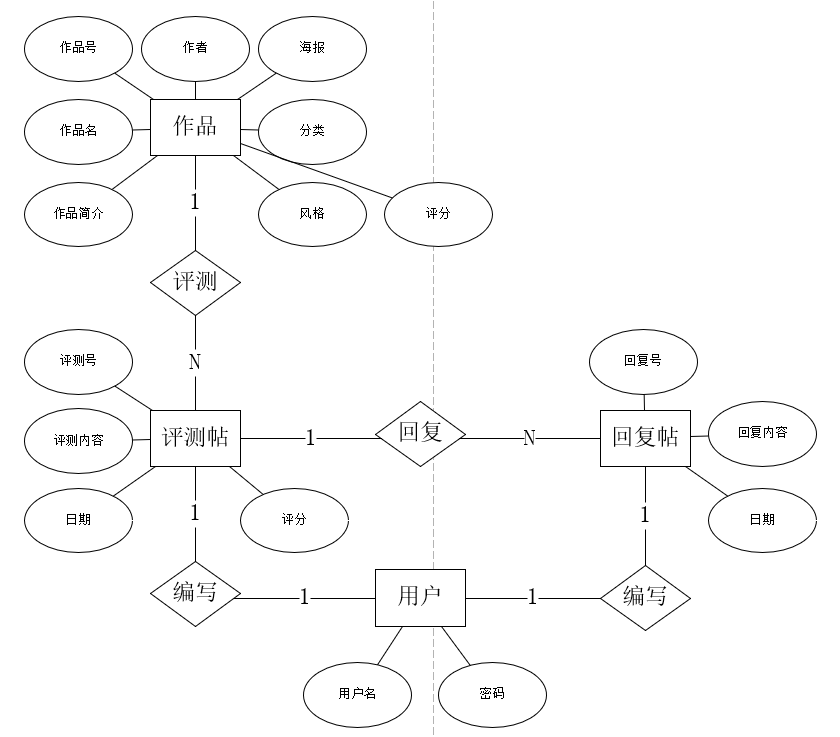
联系：

一个作品可以有多个评测帖

一个评测帖由一个用户创建，可以有多个回复帖

一个回复帖由一个用户创建，只回复一个评测帖

以下是E-R图：



* + 1. **逻辑设计**

根据E-R图进行关系模式转化得到：

作品（作品号，作品名，作品简介，作者，海报，分类，风格，评分）

评测帖（评测号，评测内容，日期，评分，作品号，用户名）

回帖子（回复号，回复内容，日期，评测号，用户名）

用户（用户名，密码）

* + 1. **物理结构设计**

根据关系模式投射到物理结构表的设计：

数据库的存储方式：服务端本地MySQL数据库，localhost:8085端口

数据库的建立方式：使用JPA提供的自动创建数据表的方式，不需要直接编写SQL语句。在项目的domain文件夹下编写JavaBean，里面设置配置就可以直接创建数据库表。

以下是每一张表（每一个Java类）的结构设计

注：每个类都提供了无参构造函数，get和set方法，使用注解确定各类约束

|  |  |  |  |
| --- | --- | --- | --- |
| 表1 作品表 | | | |
| 数据库表JavaBean类名Product | | | |
| 名称 | 类型 | 属性 | 备注 |
| id | int | Primary Key | 作品号 |
| name | varchar(255) | not null | 作品名 |
| introduction | varchar(255) |  | 作品简介 |
| author | varchar(255) |  | 作者 |
| poster | BLOB |  | 海报 |
| type | varchar(255) | not null | 分类 |
| style | varchar(255) | not null | 风格 |
| score | int | not null | 评分 |
| 表2 评测帖表 | | | |
| 数据库表JavaBean类名Comment | | | |
| 名称 | 类型 | 属性 | 备注 |
| id | int | Primary Key | 评测号 |
| context | varchar(255) | not null | 评测内容 |
| score | int | not null | 评分 |
| date | date | not null | 日期 |
| productid | int | Foreign Key | 作品号 |
| userName | varchar(255) | Foreign Key | 用户名 |
| 表3 回复表 | | | |
| 数据库表JavaBean类名Reply | | | |
| 名称 | 类型 | 属性 | 备注 |
| id | int | Primary Key | 回复号 |
| context | varchar(255) | not null | 回复内容 |
| date | date | not null | 日期 |
| commentId | int | Foreign Key | 评测号 |
| userName | varchar(255) | Foreign Key | 用户名 |
| 表4 用户表 | | | |
| 数据库表JavaBean类名User | | | |
| 名称 | 类型 | 属性 | 备注 |
| name | varchar255() | Primary Key | 用户名 |
| password | varchar(255) | not null | 密码 |

* 1. **系统设计**
     1. **概要设计**

**前端页面：**

1. 主要页面 Main.vue

实现对作品名的模糊查询

实现高分作品的摘要显示

实现精彩影评的摘要显示

实现登录和注册弹窗

实现排行榜查看

跳转综合查询页面、用户管理页面

1. 综合查询页面 Select.vue

实现对作品的分类选择

实现对作品的模糊查询

1. 作品评测页面 Product.vue

作品内容显示

作品评论显示和打分添加

评论的回复显示和添加

1. 用户浏览页面 User.vue

用户信息管理

添加作品

自己的评论浏览，删除

**后端数据库：**

1. 作品表

查找10个推荐结果，按照打分降序 selectTenRecommend

实现对作品名的模糊查询 selectAllByName

实现对作品的综合模糊查询 selectAll

实现对特定分类的排行榜查询，按照打分降序selectAllByTypeOrderByScore

实现添加作品 insertOne

1. 评测表

查找10个推荐结果，按照打分降序，selectTenRecommend

实现对相关作品号的评测查询 selectAllByProductId

实现对相关用户的评测查询 selectAllByUserName

实现添加评测 insertOne

实现删除评测 deleteOne

1. 回复表

实现对相关评测号的回复查询 selectAllByCommentId

实现添加回复 insertOne

1. 用户表

查找用户是否存在 checkByUserName

用户名密码验证 checkUser

添加用户 insertOne

修改密码 updatePassword

* + 1. **详细设计**

**配置说明：对于本项目的大体实际实现进行一下预备说明**

所有实现按照前面需求分析划分的功能模块，然后先实现数据库，再前后端并行开发调试。数据库表通过JPA在后端第一次运行后自动在MySQL中生成。

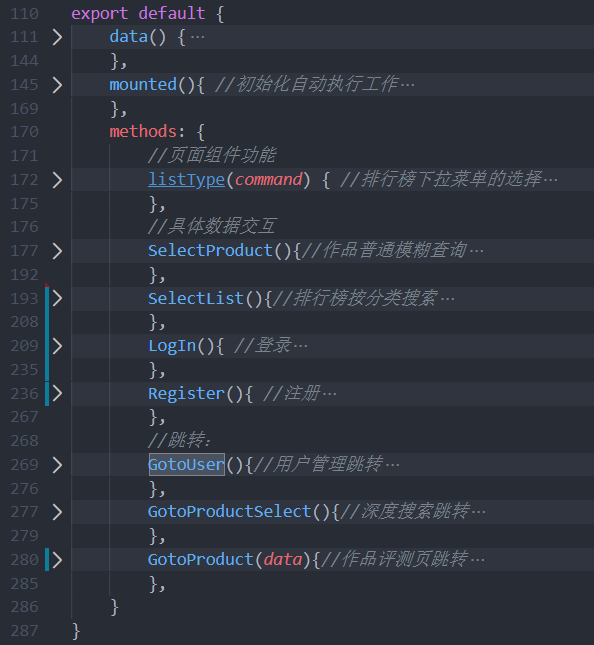
数据库端口：3306，前端：8080，后端：8085

实际的前后端实现的时候，前端依照需求简单设计了UI并辅以简单CSS；后端依照需求通过集成的SpringMVC一个个实现控制，再Service实现具体业务逻辑。前后端方法命名约束统一依照概要设计当中的命名。

**前端页面功能：（简列核心功能函数）**

Main.vue 主页面，访问的首页

mounted初始化该页面的默认评测推荐、默认内容查询、默认排行榜生成



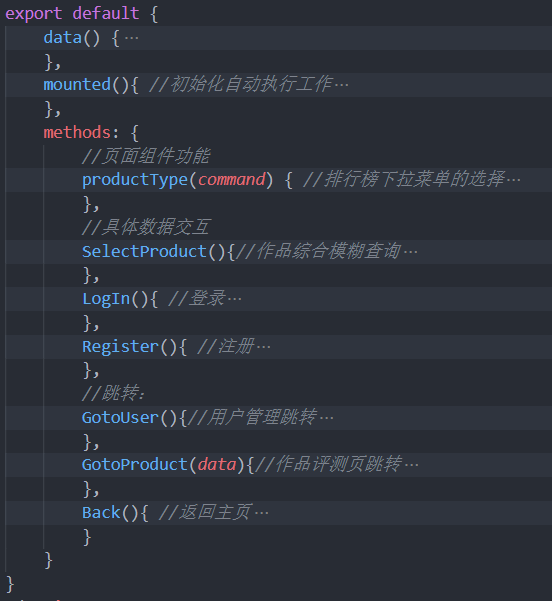
Product.vue 作品页面，显示作品信息，相关评测帖和回复帖

mounted初始化作品页面的信息读入，对应评测帖查询显示



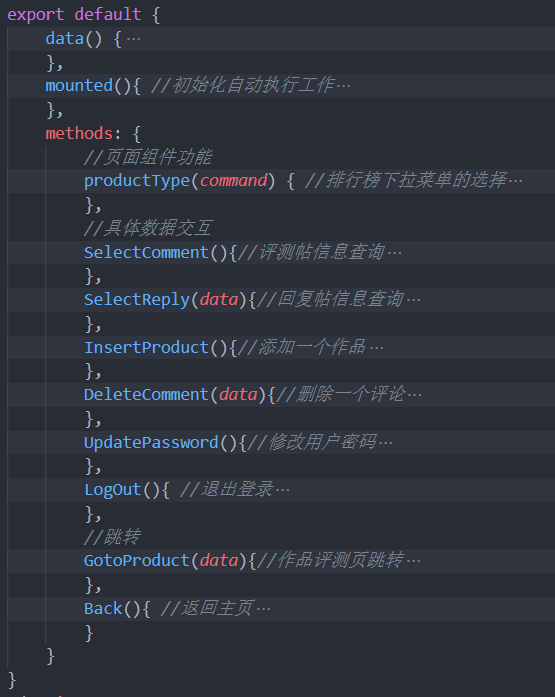
Select.vue 综合查询页面，提供多种元素的模糊匹配查询。

mounted初始化作品页面的信息读入，对应评测帖查询显示



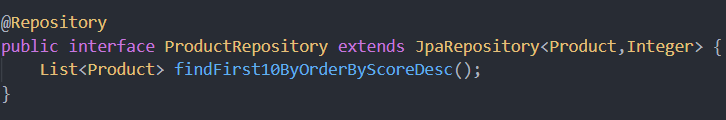
User.vue 用户页面，显示用户信息，修改密码，处理评测，添加作品

mounted初始化作品页面的信息读入，对应评测帖查询显示



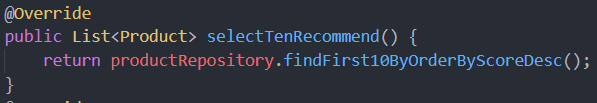
**后端数据库功能：尽量简化了控制层，专注于业务逻辑层实现功能**

作品数据操作



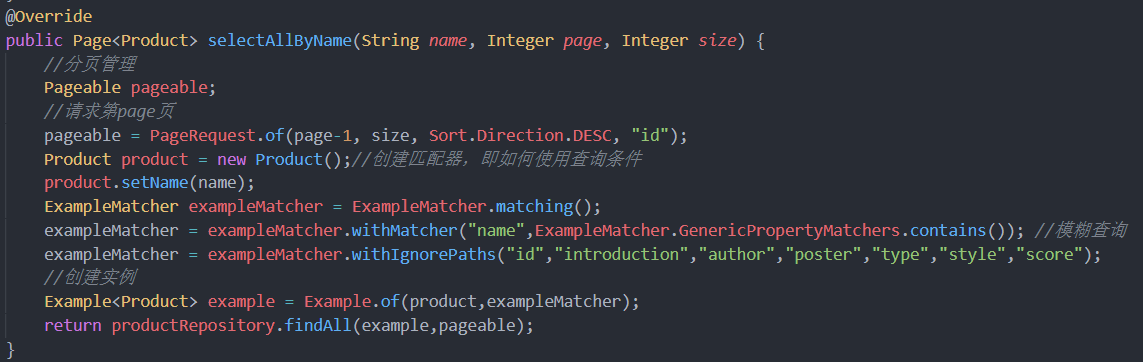
查找10个推荐结果，按照打分降序 selectTenRecommend

运用JPA语法自定义了Repository的功能函数

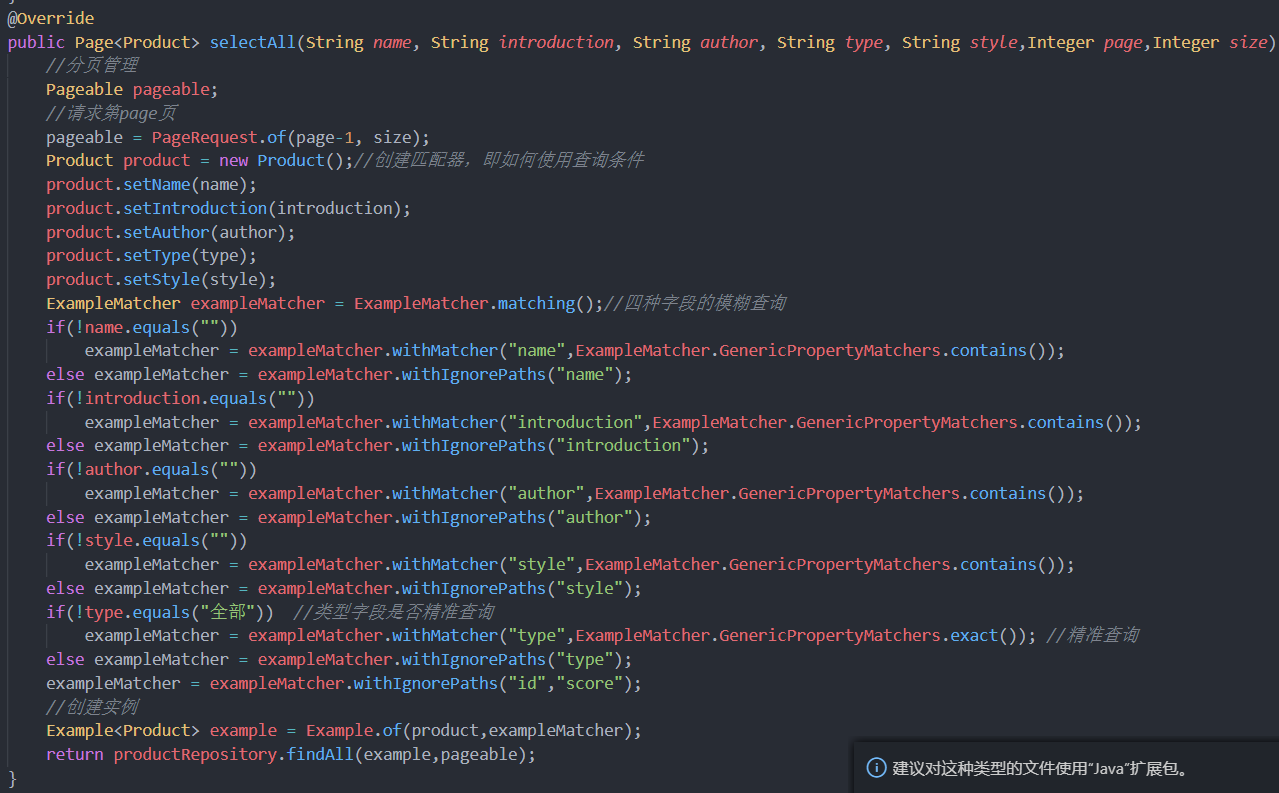


实现对作品名的模糊查询 selectAllByName

通过example和pageable来设置相关查询条件

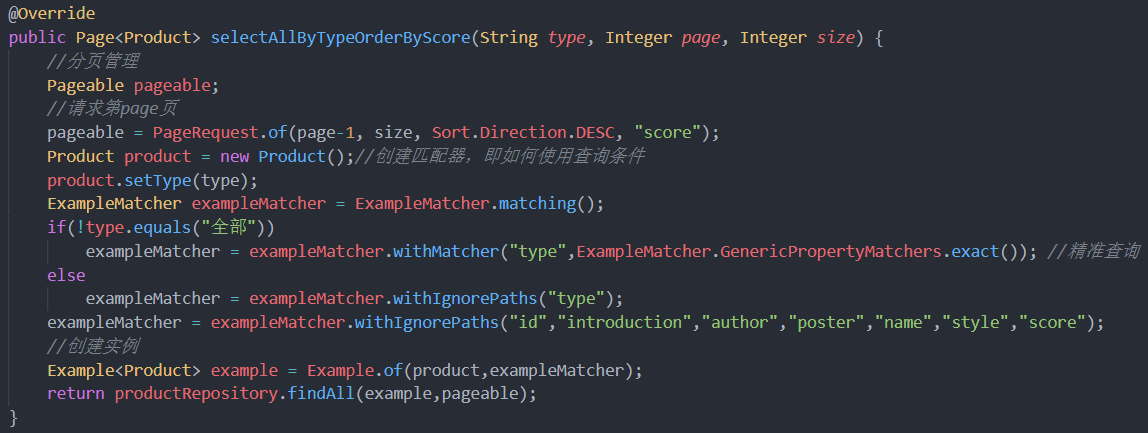


实现对作品的综合模糊查询 selectAll

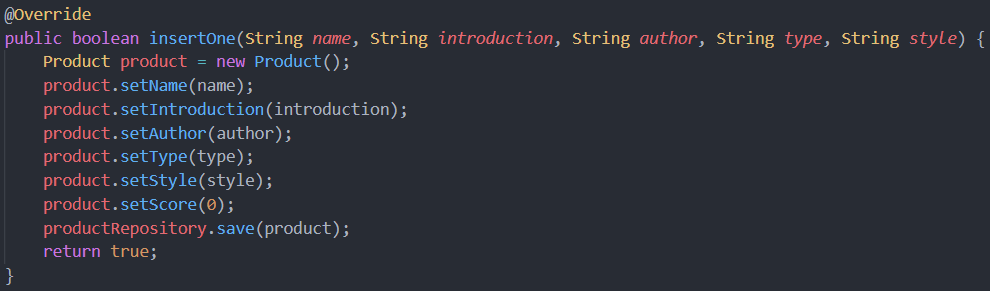


实现对特定分类的排行榜查询，按照打分降序selectAllByTypeOrderByScore

分类type一定是精准查询，不是单单做匹配。

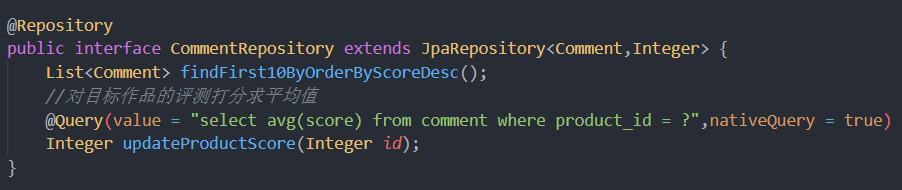


实现添加作品 insertOne

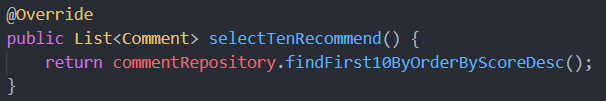


评测数据操作

原生SQL直接实现计算平均分



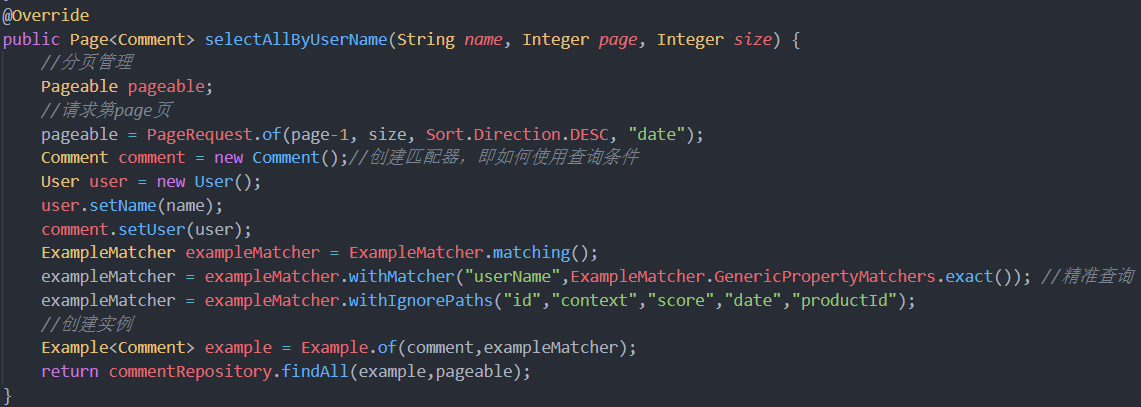
查找10个推荐结果，按照打分降序，selectTenRecommend



实现对相关作品号的评测查询 selectAllByProductId



实现对相关用户的评测查询 selectAllByUserName



实现添加评测 insertOne

日期获取当前日期

添加完毕后，调用update函数更新相关作品打分



实现删除评测 deleteOne



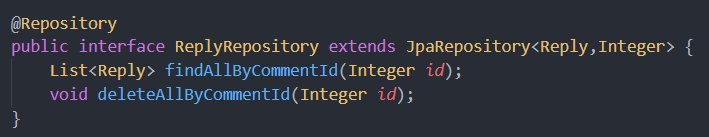
**业务逻辑：更新作品平均得分处理**

这个函数是自定义的，输入参数为要更新的作品的主键id。

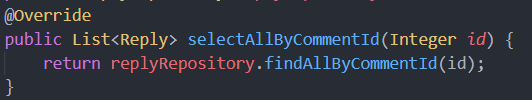
先获取到对象，然后调用数据支持求得平均分后，再重新save作品对象



回复数据操作

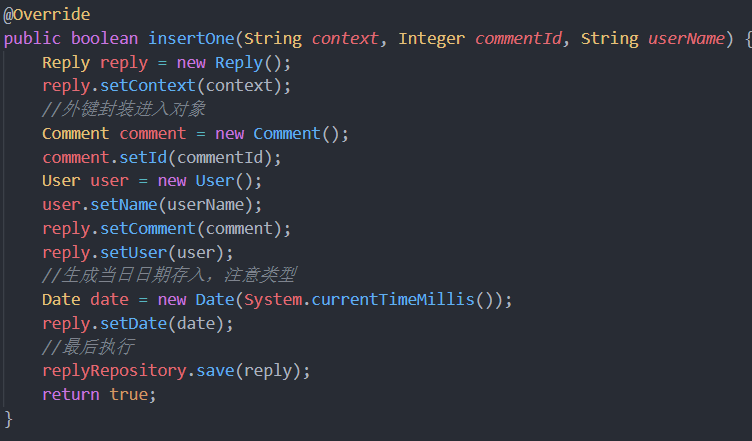


实现对相关评测号的回复查询 selectAllByCommentId

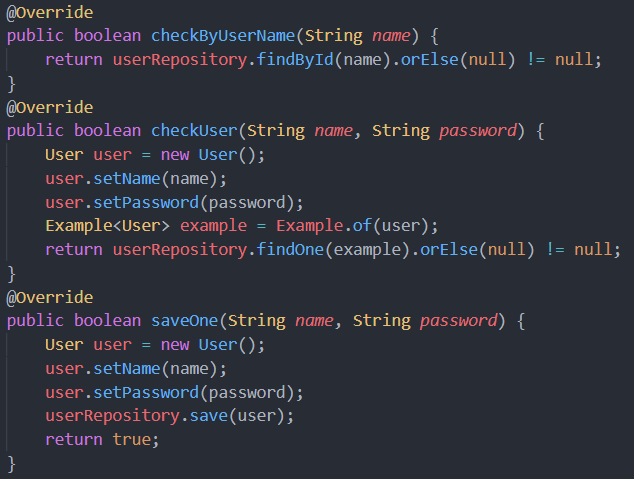


实现添加回复 insertOne

日期获取当时日期



用户数据操作 简单化设计



**其他细节说明：**

主页和作品页面均提供用户登录和注册功能，登录之后会马上刷新页面。

用户数据通过Vue-session存在前端，跨页读取和登录记忆

1. **项目测试（展示）**

具体测试内容，因为答辩的时候有演示，直接参见附件内录屏有当时现场版。

1. **项目评价**
   1. **系统不足和改进空间**

首先，这个评测论坛系统还有很多可以细化优化的部分和细节，也有待扩展的部分。所有基本的功能都是完备的，但是有些设计的不尽如人意，以及比较缺乏一些独特的地方。实现的方式并不复杂，选用的技术简单但是我觉得实用。

前端因为不是擅长部分，就作为系统页面简单完成。前后端的数据交互可以改进；后端有一些部分或许可以简化。

* 1. **自身要努力的地方**

这次不同于数据库应用课程设计，选择了独立完成因此也没想到弄个很大的项目，先前也接触了SpringBoot然后通过这门课程的学习补充了相关原理知识。于是决定做这么个比较基础但是又不像传统选题的评测论坛系统。没有用很多高大上的技术，自身注重后端开发所以前端设计的比较简单，然后这个项目的业务逻辑也不是太复杂。最终完成，以后如果从事Java开发相关的工作，还需要多用多学习，多去挖掘原理，辅以API的扩展掌握。

1. **附录**

提交说明

项目文件夹下有 PPT，综合实验报告，附录文件夹

附录文件夹下有Vue是前端工程，SpringBoot是后端工程