

# 高级webPJ文档

本次PJ我们小组实现了一个利用微信小程序的学习程序，教师端使用mvc框架+jsp实现，微信程序后端使用springboot+spring data jpa实现，总的来说基本满足了项目的要求。

## 小组成员与分工占比

成员	职务	占比	主要工作
何思良	组长	30%	微信小程序后端的搭建与服务器数据库的部署
陆昊洋	组员	30%	教师端代码的编写与修改
曹铭明	组员	30%	微信小程序前端的编写与修改
陈自博	组员	10%	微信小程序前端css（因家里有事没有做）

## 项目地址

数据库地址：---rm-bp1wu655lv304sexz.mysql.rds.aliyuncs.com:3306/onlinepj

教师端前端地址：<http://139.155.143.116:8090>

## 数据库表结构

### users

用户表，存储用户相关信息

UID	用户ID，自增，主键
username	用户账号名称
Pass	密码

### student

班级表，通过用户ID与课程ID匹配来查询选了某课程的所有用户或者查询某用户是否选择了某课程

CourseID	课程ID，关联课程表
UID	用户ID，关联用户表

### course

课程表，存储课程相关信息

CourseID	课程ID，自增，主键
coursename	课程名称
Discription	课程介绍
imagepath	课程图片路径
UID	开课老师的用户ID
createtime	开课时间

**courseuint**

章节表，存储所有章节信息

UnitID	章节ID，自增，主键
unitname	章节名称
CourseID	章节所在课程的课程ID

**coursecell**

知识点表，存储所有知识点信息

CellID	知识点ID，自增，主键
UnitID	知识点所在章节的章节ID
cellname	知识点名称
Discription	知识点介绍
Path	知识点相关文件路径

**resource**

资源表，存储教师上传的课程资源相关信息

RID	资源ID，自增，主键
CourseID	资源所在课程的课程ID
createtime	上传时间
Path	资源文件路径

**homework**

作业表，存储作业相关信息

HID	作业ID, 自增, 主键
CourseID	作业所在课程的课程ID
Discription	作业内容

### execution

作业提交表, 通过作业ID和用户ID匹配来查询作业提交情况和批改情况

HID	作业ID, 关联作业表
UID	提交作业的用户ID, 关联用户表
Score	分数, 初始默认-1(未批改), 由老师批改上传
Homework	学生提交的作业答案

### collection

课程收藏表, 通过用户ID与课程ID匹配来查询收藏了某课程的所有用户或者查询某用户是否收藏了某课程

CourseID	课程ID, 关联课程表
UID	用户ID, 关联用户表

### note

笔记表, 存储笔记的内容, 通过用户ID与课程ID来查询某课程某用户的笔记内容

CourseID	课程ID, 关联课程表
UID	用户ID, 关联用户表
Discription	笔记内容

## 代码结构

### 教师端代码结构

负责人:陆昊洋 16302010021

教师端代码使用了mvc三层框架, 编写了相关dao、servlet和jsp来实现程序

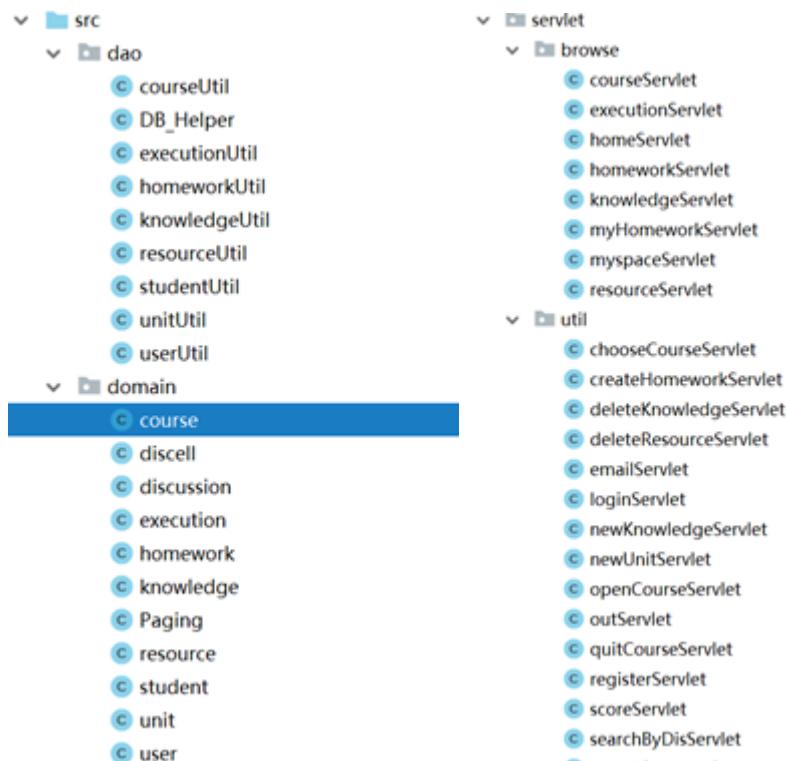
教师端程序相关jar\*\*包\*\*

```

v  lib
> commons-fileupload-1.3.2.jar
> commons-io-2.5.jar
> mail.jar
> mysql-connector-java-8.0.11.jar

```

## 教师端程序结构



Temp/msohtmlclip1/01/clip\_image006.jpg)

src文件夹下有dao文件夹，domain文件夹和servlet文件夹

dao文件夹包含了所有sql操作相关的代码

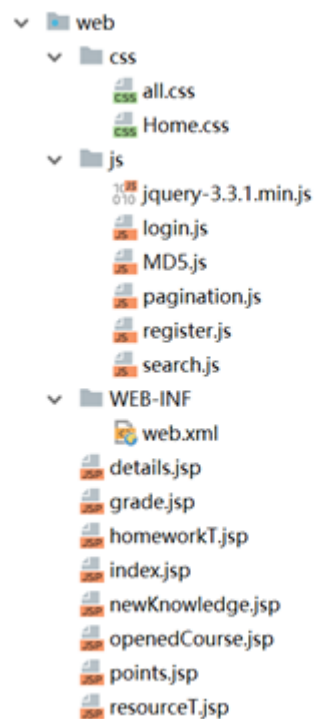
domain文件夹下包含了所有实例

servlet文件夹下的代码分两类：

browse文件夹下的servlet功能都是简单的访问，会直接跳转至具体的jsp；

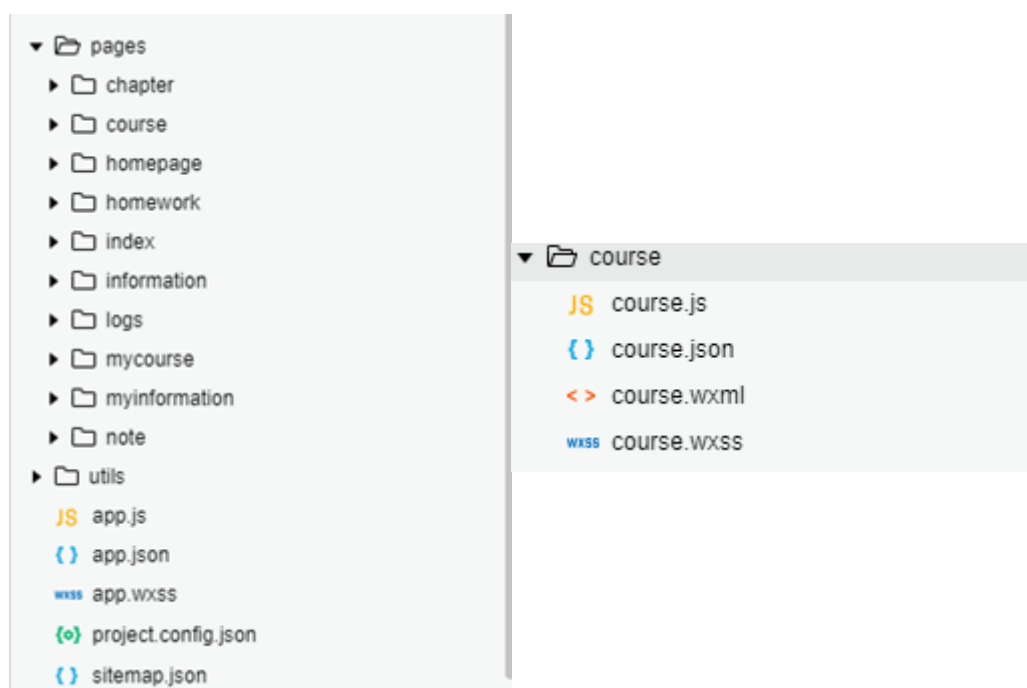
util文件夹下的servlet功能主要是进行表单的一些复杂操作，例如进行数据写入删除或者文件上传等等，之后跳转至browse内的servlet再转入具体的jsp文件。

**Web\*\*相关文件结构\*\***



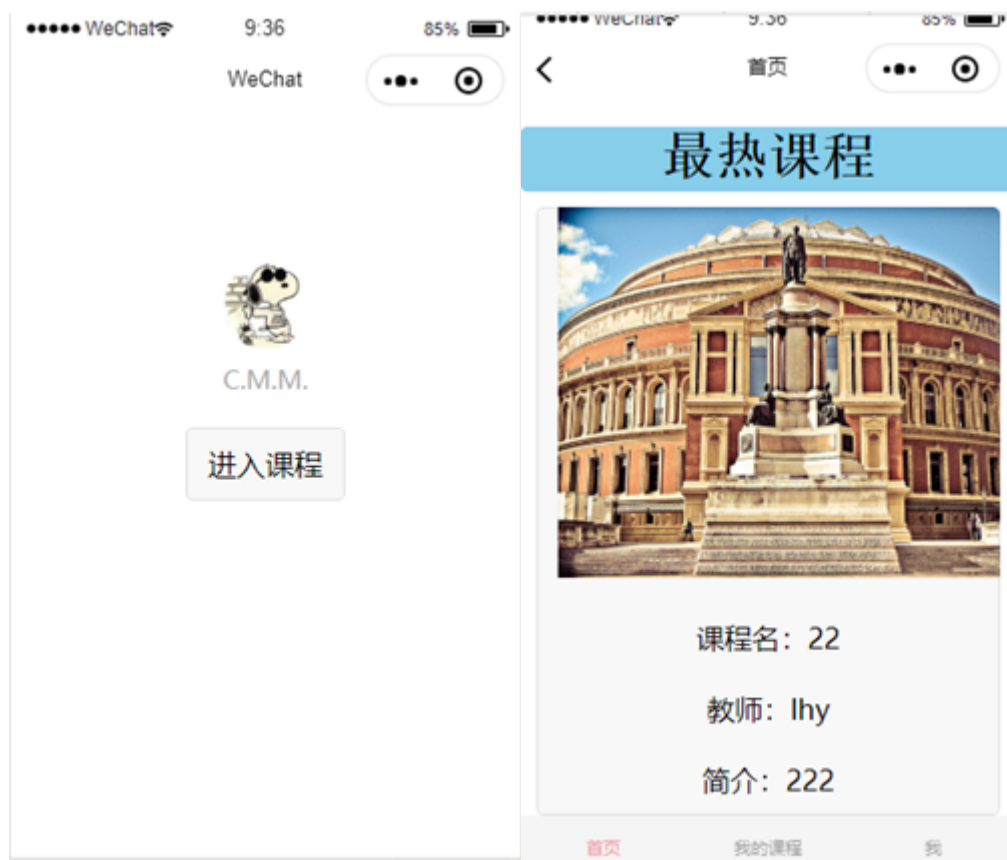
## 微信小程序项目前端结构

负责人:曹铭明 16307130079



Pages下面分别代表了不同的页面，每个页面下的结构如上图二所示，js文件用于处理各种功能的实现和数据的读取传递，json文件用于存储一些页面信息，wxml文件是页面的整体样式显示，wxss文件是定义各种组件的格式，用于美化界面

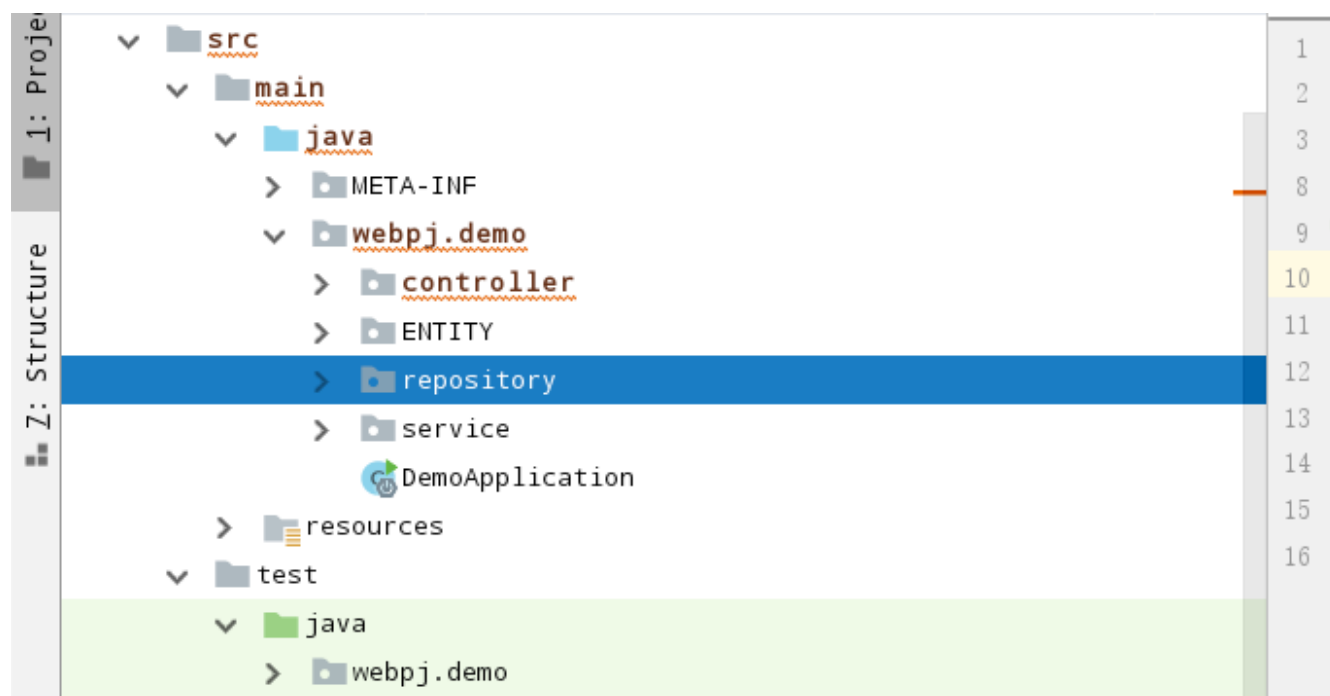
## 微信小程序运行截图



## 微信小程序后端代码结构

负责人:何思良 16030201056

大致的包结构如图



其中controller包放的是不同的controller函数，用于处理微信前端发来的url和参数，具体的函数如下图

- collectioncontroller
- coursecellcontroller
- coursecontroller
- courseunitcontroller
- Executioncontroller
- homeworkcontroller
- notecontroller
- resourcecontroller
- studentcontroller
- userscontroller

collectioncontroller是实现的用户收藏操作，包括收藏的查询等

coursecellcontroller实现课程知识点操作，coursecontroller实现查询课程等，courseunitcontroller实现课程的章节方面的操作，

Executioncontroller实现了作业的提交和批改查询操作，而homeworkcontroller存储的是老师发布的而同学未做的homework

notecontroller存放的是笔记的提交操作，studentcontroller实现的是选课操作和查询用户已选课程

userscontrller存放的是用户登录以及注册。

ENTITY存放的是实体类，而repository存放的是仓库类，所有的仓库都extends了jparepository，service里面存放具体的函数和方法，在service.impl里面实现，具体的代码结构如图

ENTITY:

- userscontroller
- ENTITY
  - CollectionEntity
  - CollectionEntityPK
  - CoursecellEntity
  - CourseEntity
  - CourseunitEntity
  - ExecutionEntity
  - ExecutionEntityPK
  - HomeworkEntity
  - NoteEntity
  - NoteEntityPK
  - ResourceEntity
  - StudentEntity
  - StudentEntityPK
  - UsersEntity

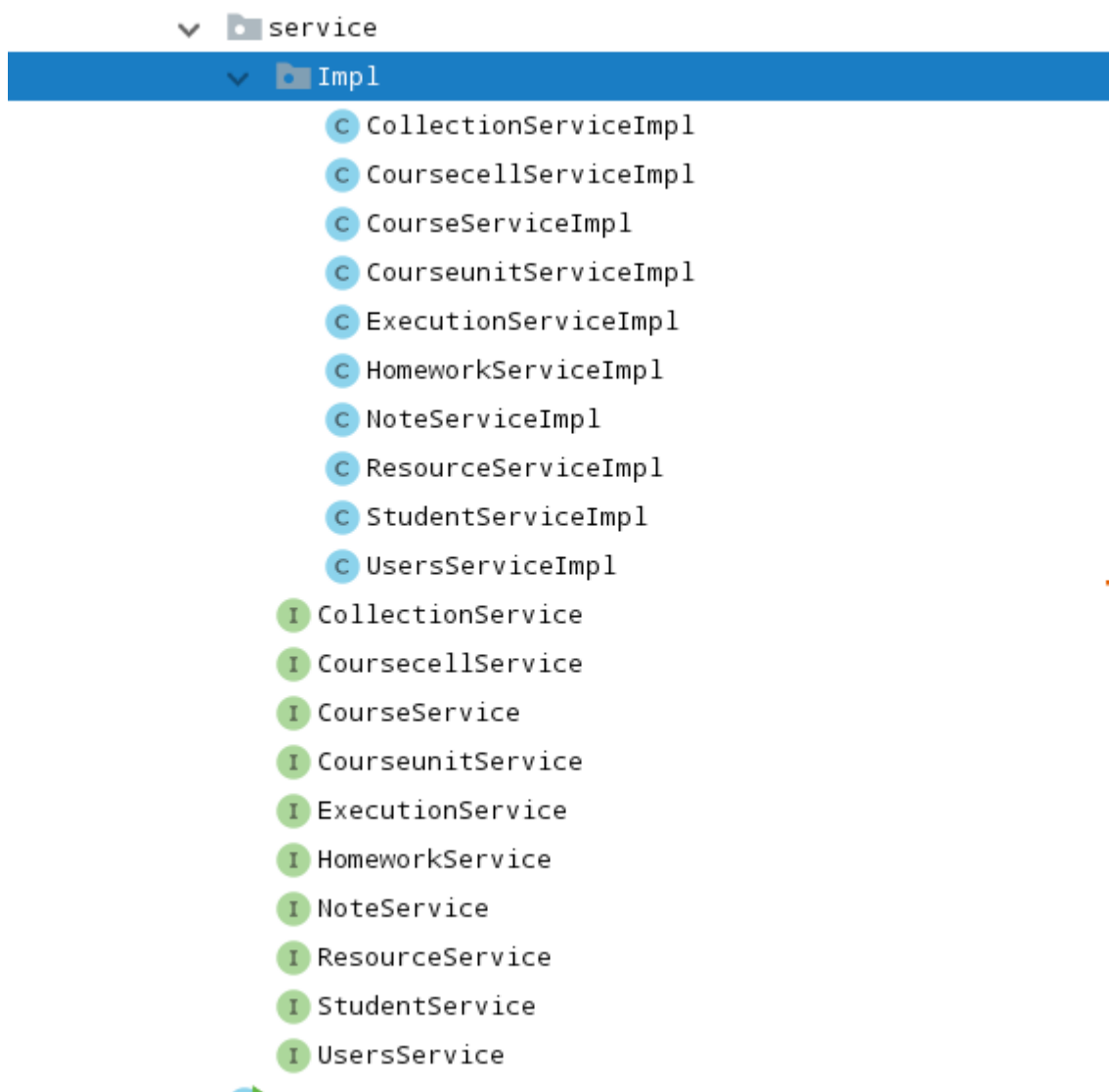
repository:

- ▼ repository
  - ▼ impl
    - CollectionRepositoryImpl
    - CoursecellRepositoryImpl
    - CourseRepositoryImpl
    - CourseunitRepositoryImpl
    - ExecutionRepositoryImpl
    - HomeworkRepositoryImpl
    - NoteRepositoryImpl
    - ResourceRepositoryImpl
    - StudentRepositoryImpl
    - UsersRepositoryImpl
  - CollectionRepository
  - CollectionRepositoryCustom
  - CoursecellRepository
  - CoursecellRepositoryCustom
  - CourseRepository
  - CourseRepositoryCustom
  - CourseunitRepository
  - CourseunitRepositoryCustom
  - ExecutionRepository
  - ExecutionRepositoryCustom
  - HomeworkRepository
  - HomeworkRepositoryCustom
  - NoteRepository
  - NoteRepositoryCustom
  - ResourceRepository
  - ResourceRepositoryCustom
  - StudentRepository
  - StudentRepositoryCustom
  - UsersRepository
  - UsersRepositoryCustom
- ▼ service

6: TODO   Spring   Terminal   Java Enterprise   9: Version

service:





然后demoapplication是主函数，启动项目，在resources里面存放了数据库的配置。

大概的结构就是这样了。

## 开发中遇到的问题

何思良：

1 在操作数据库的时候出现问题，因为我是使用的spring data jpa，对数据库操作时有问题，它默认把例如CreateTime这样的属性生成了新的属性create\_time，这样你的操作都变成了对create\_time的操作，前端的同学跟我吐槽说东西完全读不到，最后排查数据库发现了这个问题，看网上的解决方法是使用另一个东西，或者属性改成全小写createtime，这样就解决了问题，不得不说这的确是个坑。

2 对note进行添加操作的时候前端的同学向我反映无法添加note，而我检查了方法后应该是没问题的，在浏览器上输入url后提示我duplicate-entry-string1-string2-for-key-primary，这才知道是数据库问题，note表有两个主键，把其中一个不设置成主键就圆满解决了问题。

3 本来开始是打算用mybatis的，但我尝试写了一个后端放在服务器上跑的时候发现xml文件找不到，尝试了很多种办法都没办法，后面我选择了使用spring data jpa，还是很好用的，圆满解决了这个问题。

陆昊洋：

编写教师端代码时，遇到的最大的问题是文件上传的权限问题，在本地运行测试时，需要通过管理员权限打开intellij运行程序才能成功上传文件，否则会被拒绝访问；部署到服务器后这个问题还是会在某些情况下触发，不知道该如何解决这个问题。

**曹铭明：**

微信小程序的wxss没有好的参考框架，没有找到像css中的bootstrap框架可以提供模板，因此界面很难看

微信小程序在开发者工具和真机调试时都可以正常运行，但是发布到体验版没法正常运行，后来发现原因是域名的问题，IP地址没法成为微信小程序的合法地址，必须是具体的域名

## 优点

**何思良：**

我觉得最主要的优点是使用了spring data jpa了吧，大大简化了后端，就不用写那么多dao层的东西，jpa自动实现了很多的方法，用着也很快，结构相对清晰。

第二是后端和数据库分离，这样更有利于数据的安全性，同时也有利于教师端的访问，大概就是这么多了。

**陆昊洋：**

(1) 在注册账号功能里加入了邮件验证码的功能，实现了邮件的发送和验证码匹配

(2) 编写了资源上传功能，老师可以在课程内上传资源，学生可以下载教师上传的资源自行学习

(3) 在首页部分制作了最热课程的图片轮播，视觉效果上有提升

**曹铭明：**

实现了对话式教学的基础上，在老师教学的同时，学生可以同时输入内容来记录重要的信息，作为笔记内容存储，可以在课堂之外对笔记学习，在老师教学的同时，学生可以对老师布置的作业题目进行回答，做到学习和作业同步化，同时作业不限制于选择题，还可以是问答题的形式，学生在完成了题目后老师可以实现批改，批改后学生还可以得到老师批改后的反馈

实现了项目的独立性，项目实现以数据库为基础，同时实现教师端的前后端和学生端的前后端分离，教师端采用MVC框架，学生端用微信小程序+springboot来实现，项目层次十分清晰，在开发的时候也保持了相当高的独立性