

---

# Controller API 设计

FreeMarker 标准组

---

成员

周健 方俊

辛春蕾 梁依帆

面向对象分析与设计(OOAD)

厦门大学软件学院 2016 级

FreeMarker 标准组

反馈邮箱 417484588@qq.com

版本号：第二版

2018 年 12 月 16 日

# 修订记录

[illegible]

## Freemarker Controller API标准文档说明

1. Freemarker API 所有的 API 不完全由页面请求构成，除页面请求包含很多操作性请求，而操作性请求我们通过类 Restful URL 实现，以支持前端通过 Ajax 等不刷新页面的方式提交操作请求
2. API 构成元素说明：
  - AJAX：表示该 API 为操作性请求，采取类 Rest URL 实现
  - GET/POST/PUT/PATCH/DELETE：前端 HTTP 请求类型
  - data：前端应通过 HTTP Request 传输的数据，默认类型为 Request URL
  - data-type：指定前端通过 HTTP Request 传输的数据类型，如果不指明则为 Request URL
  - model：返回给前端模版引擎 Freemarker 的数据
  - return：返回值，包含两种，页面和 HTTP 状态。其中页面以 Page(页面名) 形式说明，HTTP 状态以 Response Status code 形式说明
  - exception：异常，指定当服务器处理一些请求时应当抛出的异常，也为 HTTP 状态
3. 关于Data-Type，原则上，PUT/PATCH/DELETE 等方法传递数据只有Json与Path Variable两种方法，POST 方法可适配任何传输数据类型。
4. 关于异常，API将会定义部分异常的触发，不规定所有异常的触发。比如管理员创建教师时，若提交的 DTO 有空字段等触发 BAD\_REQUEST 等，API不作要求。
5. 部分涉及到的 HTTP Response Status code 解释
  - 200：HttpStatus.OK
  - 400：HttpStatus.BAD\_REQUEST
  - 401：HttpStatus.UNAUTHORIZED
  - 403：HttpStatus.FORBIDDEN
  - 409：HttpStatus.CONFLICT
6. 对象模型与关系模型字段未一致，非一致字段请用ResultMap映射
7. 与分页相关的参考资料:[https://blog.csdn.net/qg\\_37936542/article/details/79163125](https://blog.csdn.net/qg_37936542/article/details/79163125)

## 部分类型说明

```
Team {  
    private Integer id;  
    private String serial;  
    private String teamName;  
    private String klassId;  
    private String courseId  
    private String leaderId;  
    private Integer status  
  
    private Student leader;  
    private List<Student> students;  
}
```

```
Attendance {  
    private Integer id;  
    private Integer klassSeminarId;  
    private Integer teamId  
    private Integer teamOrder  
    private boolean isPresent;  
    private String reportName;  
    private String reportUrl;  
    private int pptName;  
    private int pptUrl;  
  
    private Team team;  
}
```

```
Round {  
    private Integer id;  
    private Integer roundSerial;  
    private Integer courseId;  
  
    private List<Seminar> seminars;  
}
```

```
SeminarScore{  
    private Integer klassSeminarId;  
    private Integer teamId;  
    private Integer totalScore;  
    private Integer preScore;  
    private Integer questionScore;  
    private Integer reportScore;  
  
    private Team team;  
    private Klass klass;  
    private Seminar seminar;  
}
```

## 管理员PC端 controller API

---

```
//获取登陆界面  
GET /admin/  
GET /admin/login  
return:  
    Page("login")
```

翻转课堂管理员登录

账户名

密码

登录

//登录按钮

(AJAX)

POST /admin/login

data:

account

password

return:

@ResponseBody

认证成功: HTTPStatus(200)

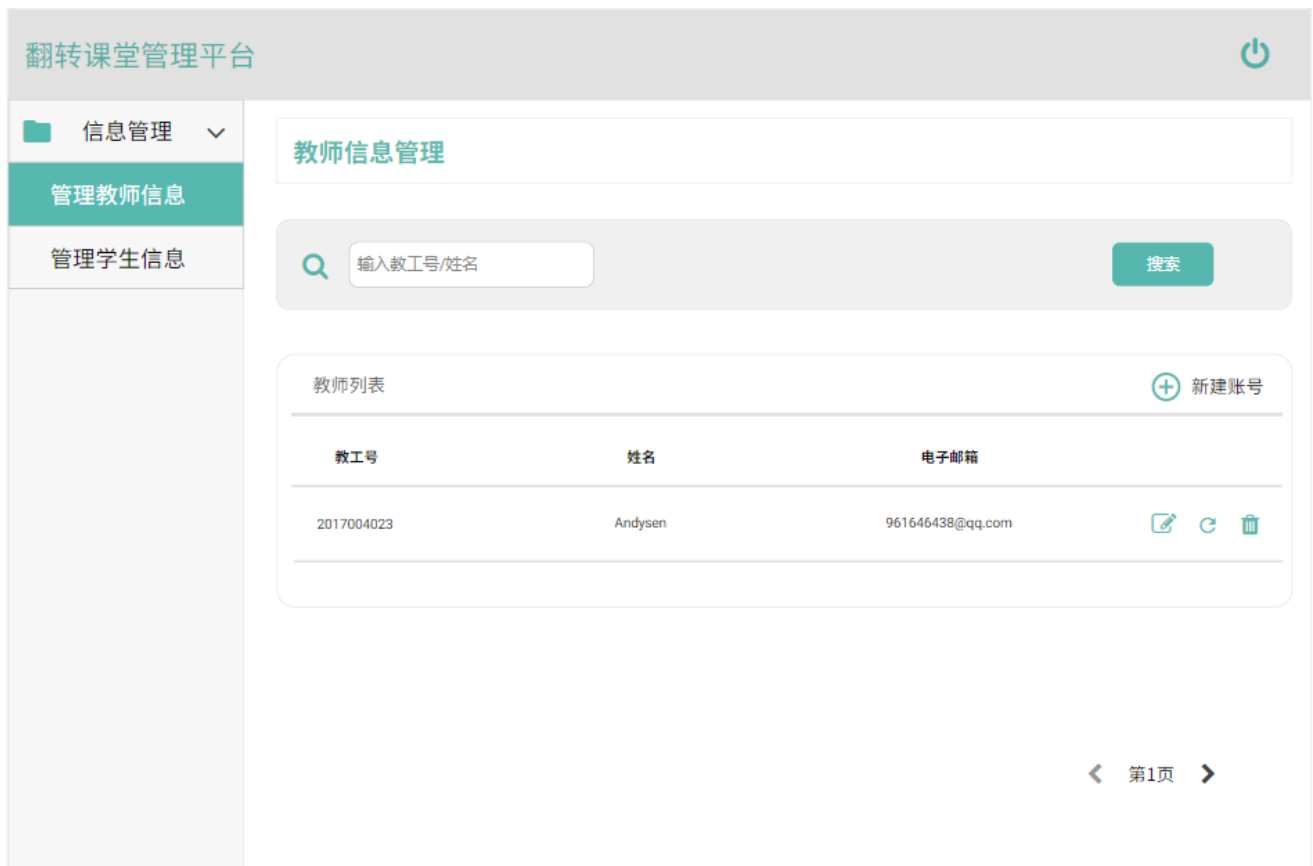
认证失败: HTTPStatus(401)

//继上一步认证成功之后

GET /admin/index

return:

Page("index")



//管理教师信息按钮

```
GET /admin/teacherListPage
return:
    Page("teacherList")
```

//加载teacherList页面时需要向服务器请求教师列表

//这边是使用bootstrapTable来提交请求并返回相应教师列表

```
GET /admin/teacherList
data:
    limit:一次加载的数量
    offset:偏移量
    keyword:用于搜索的关键字，可以为空
return:
    @ResponseBody
    PageHelper<Teacher>
```

//PageHelper类说明

```
public class PageHelper<T> {
    /**
     * 实体类集合
     */
    private List<T> rows = new ArrayList<T>();
    /**
     * 数据总条数
     */
    private int total;
}
```

```
//搜索按钮
//同样使用了bootstrapTable来提交请求并返回相应教师列表
GET /admin/teacherList
  data:
    limit:一次加载的数量
    offset:偏移量
    keyword:用于搜索的关键字
  return:
    @ResponseBody
    PageHelper<Teacher>
```

```
//新建账号按钮
GET /admin/addTeacherPage
  return:
    Page("createTeacherPage")
```

```
//修改信息按钮
GET /admin/modifyTeacherPage
  data:
    teacherId
  model:
    addAttribute("teacher", Teacher)
  return:
    Page("modifyTeacherPage")
```

```
//重置密码按钮
(AJAX)
POST /admin/teacher/{id}/resetPwd
  return:
    @ResponseBody
    重置成功:HttpStatus(200)
```

```
//删除教师按钮
(AJAX)
DELETE /admin/teacher/{id}
  return:
    @ResponseBody
    删除成功:HttpStatus(200)
```

翻转课堂管理平台

信息管理

管理教师信息

管理学生信息

创建教师用户

教师姓名：

请输入教师姓名

教工号：

请输入教工号

电子邮箱：

请输入教师电子邮箱

初始密码：

请输入初始默认密码

提交

取消

//提交按钮

( AJAX )

PUT /admin/teacher

data:

account

teacherName

email

password

return:

@ResponseBody

创建成功:HttpStatus(200)

创建失败:HttpStatus(409)





信息管理

管理教师信息

管理学生信息

### 修改教师信息

教师姓名	<input type="text" value="李四"/>
教工号	<input type="text" value="10000086"/>
电子邮箱	<input type="text" value="123@163.com"/>

修改

取消

//修改按钮

(AJAX)

PATCH /admin/teahcer

data:

id

account

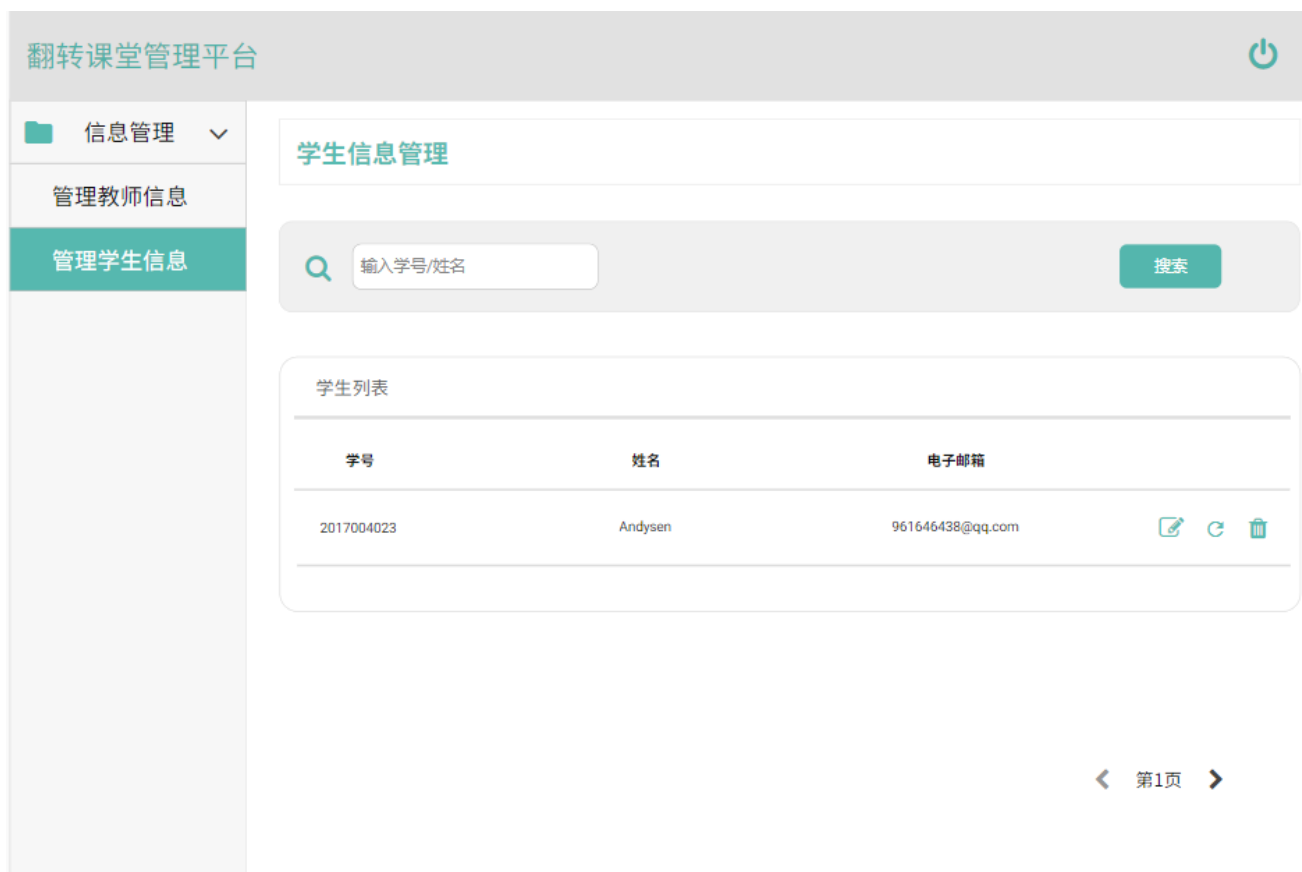
teacherName

email

return:

@ResponseBody

修改成功:HttpStatus(200)



//管理学生信息按钮

```
GET /admin/studentListPage
return:
    Page("studentList")
```

//加载studentList页面时需要向服务器请求学生列表  
//加载方式同教师端


```
GET /admin/studentList
data:
    limit:一次加载的数量
    offset:偏移量
return:
    @ResponseBody
    PageHelper<Student>
```


//修改信息按钮

```
GET /admin/modifyStudentPage
data:
    studentId
model:
    addAttribute("teacher", Student)
return:
    Page("modifyStudentPage")
```

```
//重置密码按钮
(AJAX)
POST /admin/student/{id}/resetPwd
return:
    @ResponseBody
    重置成功:HTTPStatus(200)
```

```
//删除学生按钮
(AJAX)
DELETE /admin/student/{id}
return:
    @ResponseBody
    删除成功:HTTPStatus(200)
```

翻转课堂管理平台 

信息管理 

管理教师信息

管理学生信息

修改学生信息

学生姓名:

学号:

电子邮箱:

修改

取消

```
//修改按钮
(AJAX)
PATCH /admin/student
data:
    id
    account
    studentName
    email
return:
    @ResponseBody
    修改成功:HTTPStatus(200)
```

# 教师PC端 controller API

//获取登陆界面

GET /pc/login

return:

Page("PcLoginPage")

## 翻转课堂管理平台

### 翻转课堂教师登录

账户名

账户密码

登录

//登录按钮

(AJAX)

POST /pc/login

data:

account

password

return:

@ResponseBody

认证成功: HTTPStatus(200+"身份信息")

认证失败: HTTPStatus(401)

//继上一步登录成功之后

GET /teacherpc/index

model:

addAttribute("courseList",List<Course>)

或者使用Map来存所需要的数据即可，举例如下：

教师主界面仅需要course的name和course的id

```
List<Map<String,Object>> list
```

```
for(...)
```

```
{
```

```
    Map m=new HashMap();
```

```
    m.put("courseName",name);
```

```
    m.put("courseId",id);
```

```
    list.add(m);
```

```
}
```

```
addAttribute("courseList",list)
```

对之后的controller API同样适用

```
return:
```

```
Page("teacherIndex")
```

## 个人信息综合管理平台



选择课程

OOAD

进入

J2EE

进入

软件工程

进入

.Net

进入

//进入按钮

POST /teacherpc/course

data:

courseId

```
return:
```

```
Page("coursePage")
```

导入学生名单

讨论课

导出成绩

## OOAD——导入学生名单

### 班级列表

1班	2016_1.xlsx	提交	重置
2班	<input type="button" value="Choose File"/> No file chosen	提交	重置
3班	<input type="button" value="Choose File"/> No file chosen	提交	重置

//矩形部分采用iframe

//导入学生名单按钮

```
POST /teacherpc/course/importStudent
data:
  courseId
model:
  addAttribute("classList",List<Klass>)
return:
  Page("importStudentPage")
```

//讨论课按钮

```
POST /teacherpc/course/seminar
data:
  courseId
model:
  addAttribute("roundAndSeminarList",List<Round>)
return:
  Page("seminarPage");
```

//导出成绩按钮

```
POST /teacherpc/course/exportScore
data:
  courseId
model:
  addAttribute("scoreList",List<Map<String,Object>>)
```

说明：

```
List<Map<String,Object>> list
for(...)
{
    Map m=new HashMap();
    m.put("teamSerial",String);
    m.put("seminarName",String);
    m.put("score",SeminarScore);
    list.add(m);
}
return:
Page("scorePage")
```

## OOAD——导入学生名单

### 班级列表

1班	2016_1.xlsx	提交	重置
2班	<input type="button" value="Choose File"/> No file chosen	提交	重置
3班	<input type="button" value="Choose File"/> No file chosen	提交	重置

//提交按钮

(AJAX)

POST /teacherpc/course/importStudent/submit

data:

classId

file(文件)

return:

@ResponseBody

上传成功:HTTPStatus(200)

上传失败:HTTPStatus(400)

```
//重置按钮
(AJAX)
POST /teacherpc/course/importStudent/reset
data:
  classId
return:
  @ResponseBody
  重置成功:HTTPStatus(200)
```

## OOAD——讨论课

### 第一轮讨论课

第一次讨论课——需求分析说明书

进入

第二次讨论课——界面原型说明书

进入

第三次讨论课——界面原型说明书

进入

### 第二轮讨论课

第一次讨论课——需求分析说明书

进入

第二次讨论课——界面原型说明书

进入

第三次讨论课——界面原型说明书

进入

```
//进入按钮
POST /teacherpc/course/seminar/info
data:
  seminarId
model:
  addAttribute("seminarAndTeamList",List<Map<String,Object>>)
  说明:
  List<Map<String,Object>> list
  for(...)
  {
    Map m=new HashMap();
    m.put("curSeminar",Seminar);
    m.put("teamList",List<Team>);
```



```
        list.add(m);
    }
    return:
        Page("seminarInfo")
```

## OOAD——第一轮讨论课——第一次讨论课

主题：需求分析说明书

报名情况：2/6

报名起止日期：2018.09.01~2018.09.06

内容： 本次讨论课一共可容纳6个小组，每个小组15分钟，其中10分钟陈述，5分钟讨论。

### 已报名小组

展示次序	小组序号	组长	展示材料
1	1-1	Andysen	未上传
2			
3	1-2	Andysen	<a href="#">下载</a>
4			
5			
6			

//下载按钮

POST /teacherpc/course/seminar/info/download

data:

filename

return:

@ResponseBody

请求成功:HTTPStatus(200)

请求失败:HTTPStatus(400)

## 学生PC端 controller API

翻转课堂登录

学号/教工号

登录密码

登录

//登录按钮

(AJAX)

POST /pc/login

data:

account

password

return:

@ResponseBody

认证成功: HTTPStatus(200+"身份信息")

认证失败: HTTPStatus(401)

//继上一步登录成功之后

GET /studentpc/index

model:

addAttribute("courseList",List<Course>)

return:

Page("studentIndex")

选择课程

OOAD	进入
J2EE	进入
软件工程	进入
.Net	进入

```
//进入按钮
POST /studentpc/course
data:
  courseId
return:
  Page("coursePage")
```

选择课程

讨论课

查看成绩

## OOAD——讨论课

## 第一轮讨论课

第一次讨论课——需求分析说明书

进入

第二次讨论课——界面原型说明书

进入

第三次讨论课——界面原型说明书

进入

## 第二轮讨论课

第一次讨论课——需求分析说明书

进入

第二次讨论课——界面原型说明书

进入

第三次讨论课——界面原型说明书

进入

//讨论课按钮

POST /studentpc/course/seminar

data:

courseId

model:

addAttribute("roundAndSeminarList",List&lt;Round&gt;)

return:

Page("seminarPage");

//查看成绩按钮

POST /studentpc/course/exportScore

data:

courseId

model:

addAttribute("scoreList",List&lt;Map&lt;String,Object&gt;&gt;)

说明:

List&lt;Map&lt;String,Object&gt;&gt; list

for(...)

{

Map m=new HashMap();

m.put("roundSerial",Integer);

m.put("score",SeminarScore);

list.add(m);

}

return:

Page("scorePage")

//进入按钮

POST /studentpc/course/seminar/info

data:

seminarId

model:

addAttribute("seminarAndTeamList", List<Map<String, Object>>)

说明:

List<Map<String, Object>> list

for(...)

{

Map m=new HashMap();

m.put("curSeminar", Seminar);

m.put("teamList", List<Team>);

list.add(m);

}

return:

Page("seminarInfo")

## OOAD——第一轮讨论课——第一次讨论课

主题：需求分析说明书

报名情况：2/6

报名起止日期：2018.09.01~2018.09.06

内容：本次讨论课一共可容纳6个小组，每个小组15分钟，其中10分钟陈述，5分钟讨论。

### 已报名小组

展示次序	小组序号	组长	展示材料
1	1-1	Andysen	未上传
2			<a href="#">报名</a>
3	1-2	Andysen	已上传
4			<a href="#">报名</a>
5			<a href="#">报名</a>
6			<a href="#">报名</a>

```
//报名按钮
(AJAX)
POST /studentpc/course/seminar/info/register
data:
  teamId
  seminarId
  order
return:
  @ResponseBody
  报名成功:HTTPStatus(200)
```

## OOAD——第一轮讨论课——第一次讨论课

主题：需求分析说明书

报名情况：3/6

报名起止日期：2018.09.01~2018.09.06

内容： 本次讨论课一共可容纳6个小组，每个小组15分钟，其中10分钟陈述，5分钟讨论。

### 已报名小组

展示次序	小组序号	组长	展示材料
1	1-1	Andysen	未上传
2	1-3	Andysen	<input type="button" value="上传"/> <input type="button" value="取消报名"/>
3	1-2	Andysen	未上传
4			
5			
6			

```
//取消报名
(AJAX)
POST /studentpc/course/seminar/info/cancel
data:
  teamId
  seminarId
return:
  @ResponseBody
  取消成功:HTTPStatus(200)
  取消失败:HTTPStatus(409)
```

```
//上传按钮
(AJAX)
POST /studentpc/course/seminar/info/upload
data:
  seminarId
  teamId
  file(文件)
return:
  @ResponseBody
  上传成功:HTTPStatus(200)
  上传失败:HTTPStatus(400)
```

## OOAD——第一轮讨论课——第一次讨论课

主题：需求分析说明书

报名情况：6/6

报名起止日期：2018.09.01~2018.09.06

内容： 本次讨论课一共可容纳6个小组，每个小组15分钟，其中10分钟陈述，5分钟讨论。

### 已报名小组

展示次序	小组序号	组长	展示材料
1	1-1	Andysen	未上传
2	1-3	Andysen	<a href="#">下载</a>
3	1-2	Andysen	<a href="#">下载</a>
4	1-4	Andysen	<a href="#">下载</a>
5	1-5	Andysen	<a href="#">下载</a>
6	1-6	Andysen	<a href="#">下载</a>

```
//下载按钮
(AJAX)
POST /studentpc/course/seminar/info/download
data:
  teamId
  seminarId
  fileName
return:
  @ResponseBody
  下载成功:HTTPStatus(200)
  下载失败:HTTPStatus(400)
```

# 学生移动端 controller API

//获取登录界面

GET /login

return:

Page("login")



讨论课管理系统登录

学号/教工号

登录密码

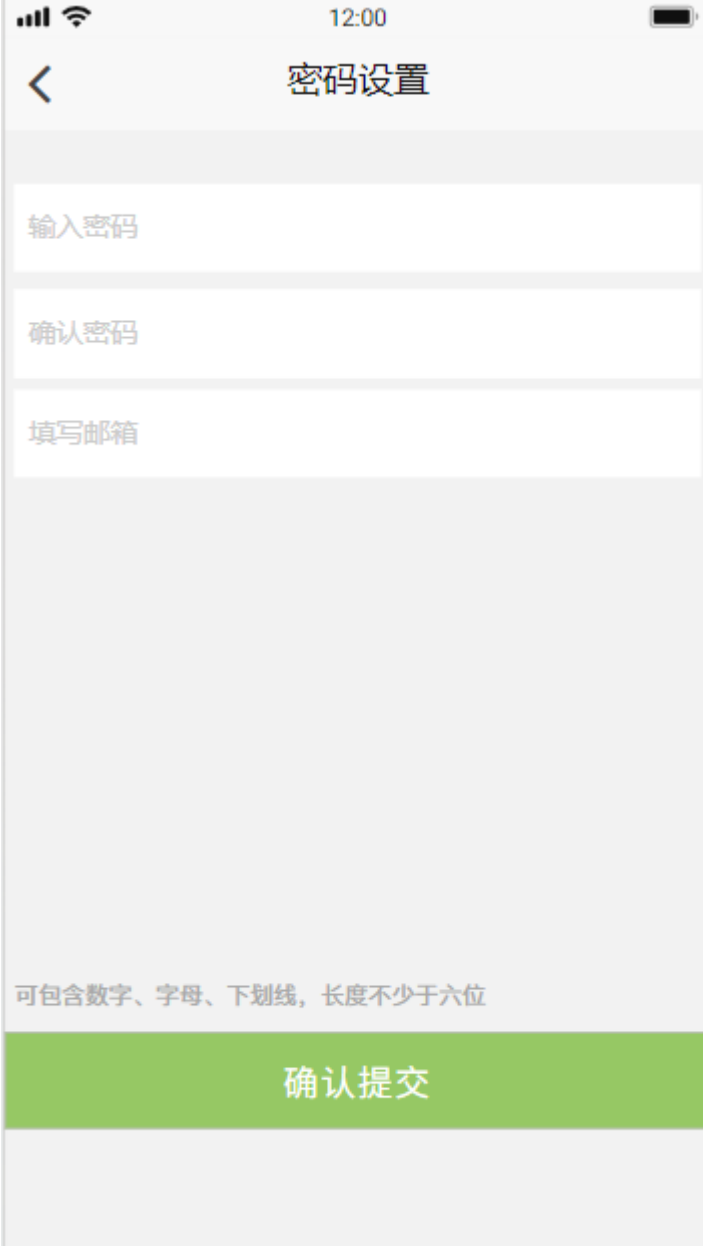
登录

忘记密码

初次登录默认密码为123456



```
//登陆按钮
(AJAX)
POST /login
data:
  account
  password
return:
  @ResponseBody
  认证成功: HTTPStatus(200+"身份信息")
  认证失败: HTTPStatus(401)
```

A mobile application interface for setting a password. At the top, there's a status bar with signal strength, Wi-Fi, and the time 12:00. Below that is a navigation bar with a back arrow on the left and the title '密码设置' (Password Setting) in the center. The main content area contains three input fields: '输入密码' (Enter Password), '确认密码' (Confirm Password), and '填写邮箱' (Enter Email). Below these fields is a large text area with a hint: '可包含数字、字母、下划线, 长度不少于六位' (Can contain numbers, letters, underscores, length not less than 6 digits). At the bottom of the form is a green button labeled '确认提交' (Confirm Submit).

```
//获取账号激活界面
GET /student/activation
return:
  Page("activation")
```

```
//确认提交按钮
POST /teacher/activation
  data:
    password
    email
  return:
    redirect:"/student/index"
```

```
//获取学生主界面
GET /student/index
  if(账号未激活)
    redirect:"/student/activation"
  else
    model :
      addAttribute("curStudent",Student)
    return:
      Page("StudentIndex")
```



//我的课程按钮

GET /student/course

model:

addAttribute("courseAndKlassList", List<Map<String, Object>>)

说明:

List<Map<String, Object>> list

for(...)

{

Map m=new HashMap();

m.put("course", course);

m.put("klass", klass);

list.add(m);

}

return:

Page("MyCoursePage")

//账户与设置

GET /student/setting

model:

addAttribute("curStudent",Student)

addAttribute("adminEmail",String)

return:

Page("setting")

The image shows a mobile application interface for the '账户与设置' (Account and Settings) page. At the top, there's a status bar with signal strength, Wi-Fi, and battery icons, and the time 12:00. Below the status bar is a navigation bar with a back arrow, the title '账户与设置', and a plus icon. The main content area consists of several rows of user information, each with a label and a value, separated by alternating light gray and white backgrounds. The information includes: 姓名 (Name): 王二 (Wang Er); 学号 (Student ID): 24320162208818; 性别 (Gender): 女 (Female); 出生年月 (Date of Birth): 1998.1; 联系方式 (邮箱) (Contact Information (Email)): 123456@qq.com, with a green '修改' (Modify) link to its right; 账户密码 (Account Password): [redacted], with a right arrow; and 管理员邮箱 (Admin Email): 654321@163.com. At the bottom, there is a large red button with the text '退出登录' (Logout).

姓名:	王二
学号:	24320162208818
性别:	女
出生年月:	1998.1
联系方式 (邮箱):	123456@qq.com <a href="#">修改</a>
账户密码	>
管理员邮箱:	654321@163.com

退出登录

//修改按钮

GET /student/setting/modifyEmail

return:

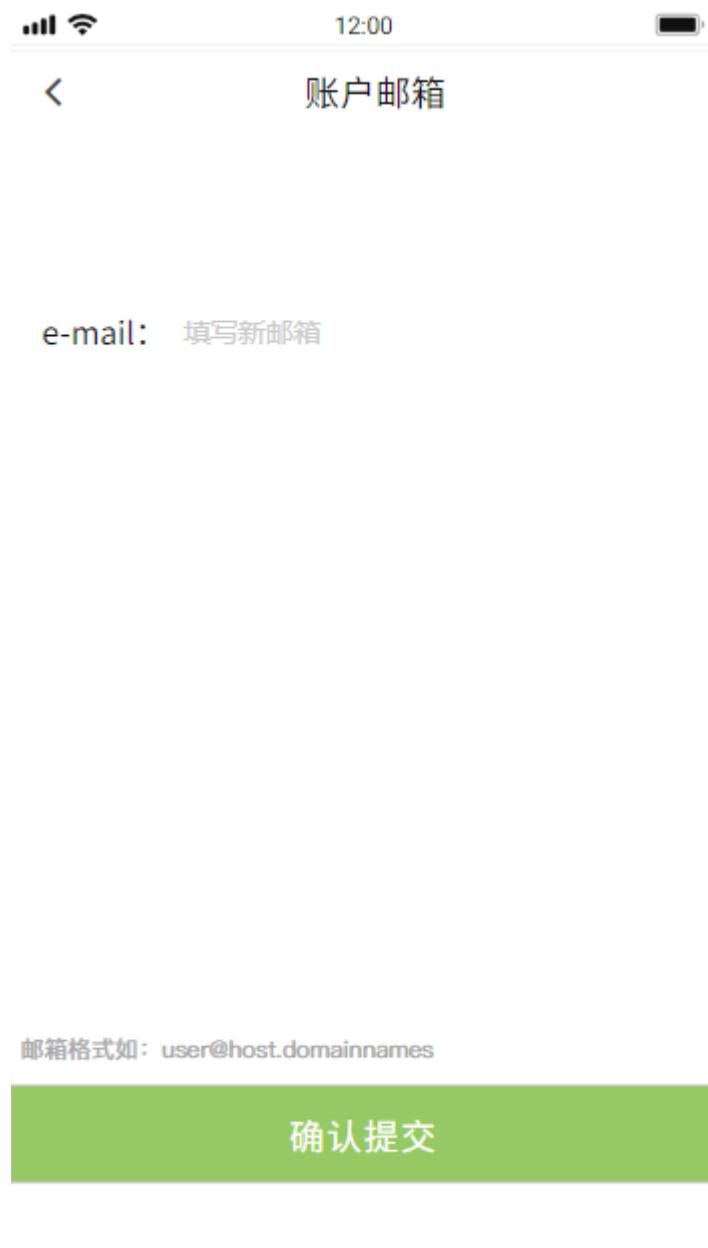
Page("modifyEmailPage")

//账户密码按钮

```
GET /student/setting/modifyPwd
```

```
return:
```

```
Page("modifyPwdPage")
```



12:00

< 账户邮箱

e-mail: 填写新邮箱

邮箱格式如: user@host.domainnames

确认提交

//确认提交按钮

(AJAX)

```
POST /student/setting/modifyEmail
```

```
data:
```

```
studentNum
```

```
email
```

```
return:
```

```
@ResponseBody
```

```
修改成功:HttpStatus(200)
```

12:00

< 账户密码

新密码： 填写新密码

确认密码： 再次填写确认

密码长度8-32位，须包含数字、字母、符号至少2种或以上元素。

确认提交

//确认提交按钮

(AJAX)

POST /student/setting/modifyPwd

data:

studentNum

password

return:

@ResponseBody

修改成功:HttpStatus(200)



//课程信息按钮

```
POST /student/course/info
data:
  courseId
model:
  addAttribute("curCourse",Course)
return:
  Page("courseInfo")
```

//我的成绩按钮

```
POST /student/course/score
data:
  courseId
  klassId
model:
  addAttribute("scoreDetails",List<Map<String,Object>>)
```

```

说明:
List<Map<String,Object>> list
for(...)
{
    Map m=new HashMap();
    m.put("roundName",String);
    m.put("seminars",List<SeminarScore>);
}
return:
Page("scorePage")

```

//我的组队按钮

POST /student/course/team

data:

courseId

model:

addAttribute("teamList",List<Team>)

addAttribute("studentsNotInTeam",List<Student>)

return:

Page("myTeam")





//我的小组按钮

POST /student/course/team/myTeam

model:

addAttribute("myTeam", Team)

return:

Page("myTeam")



//退组按钮

(AJAX)

POST /student/course/team/myTeam/quit

data:

studentId

teamId

return:

@ResponseBody

退出成功:HTTPStatus(200)

12:00

< OOAD 2016(1)

小组名: 填写组名

选择班级: 2016- ( 1 )

添加成员:

<input type="checkbox"/>	22222222222222	赵四
<input type="checkbox"/>	22222222222223	~~~
<input checked="" type="checkbox"/>	22222222222224	~~~
<input type="checkbox"/>	88851888818233	燕小六
<input type="checkbox"/>	22211122222223	~~~
<input checked="" type="checkbox"/>	22222221112224	~~~
<input checked="" type="checkbox"/>	22222222222114	~~~

确认提交

//确认提交按钮

(AJAX)

POST /student/course/team/create

data:

teamName

classId

List<studentNum>

return:

@ResponseBody

创建成功:HTTPStatus(200)

12:00

< OOAD 2016(1)

1-1 武林盟主

组长:	11111111111111	我
组员:	22222222222222	王二
	33333333333333	~~~
	44444444444444	~~~

添加成员:

<input type="checkbox"/>	22222222222222	赵四
<input type="checkbox"/>	22222222222223	~~~
<input checked="" type="checkbox"/>	22222222222224	~~~
<input type="checkbox"/>	33333333333333	王五
<input type="checkbox"/>	55555555555555	~~~

搜索成员:

解散小组

添加

保存

//组长删除成员

(AJAX)

delete /student/course/team/delete

data:

teamId

studentNum

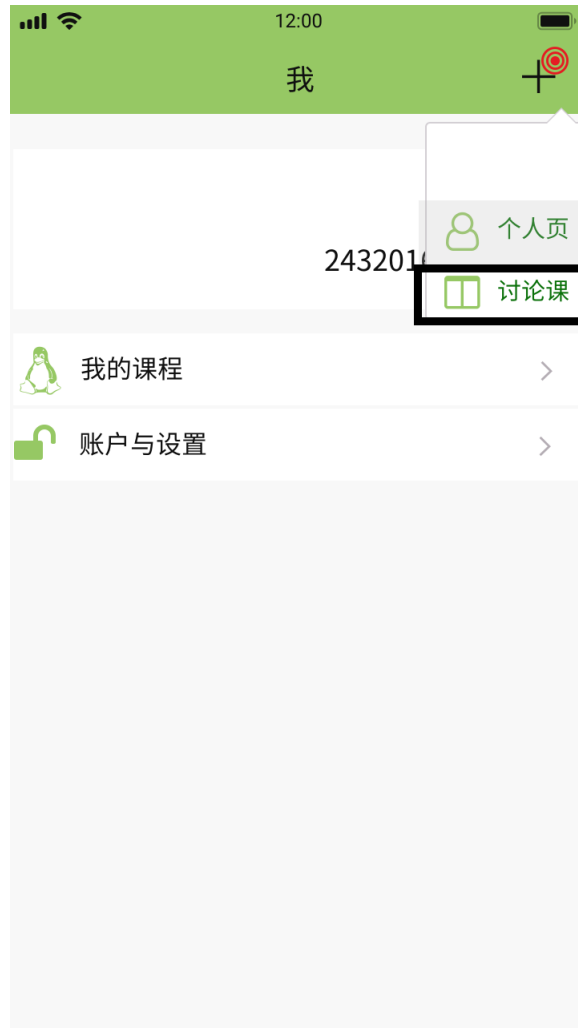
return:

删除成功:HTTPStatus(200)

```
//组长添加成员
(AJAX)
post /student/course/team/add
  data:
    teamId
    List<studentNum>
  return:
    添加成功:HTTPStatus(200)
```

```
//组长解散小组
(AJAX)
delete /student/course/team/disband
  data:
    teamId
    List<studentNum>
  return:
    成功:HTTPStatus(200)
```

```
//搜索成员
(AJAX)
GET /student/course/team/search
  data:
    keyword(学号或姓名)
  return:
    @ResponseBody
    Student
```



// 讨论课入口页

GET /student/seminarEntrance

model:

addAttribute("courseAndKlassList", List<Map<String, Object>())

说明:

List<Map<String, Object> list;

for(...)

{

Map m=new HashMap();

m.put("course", course);

m.put("klass", klass);

list.add(m);

}

return:

Page("student/seminarEntrance")



```
GET /student/course/seminarList // 讨论课列表页
data:
  courseID
  klassID
model:
  addAttribute("roundAndSeminarList",List<Round>)
return:
  Page("student/course/seminarList")
```



```
GET /teacher/course/seminar/info // 讨论课详情页
data:
  klassID
  seminarID
model:
  courseName:String
  klassSeminar:klassSeminar
  attendance:Attendance
return
Page("student/course/seminar/info")
```

12:00

< OOAD-讨论课 +

轮次：

第一轮

主题：

代码检查

课次序号：

第2次

要求：

界面导航图和所有界面原型设计  
课堂讨论每个小组15分钟

课程情况：

未开始

报名情况

报名开始时间：

10.1.2018 12:00

报名截止时间：

10.7.2018 24:00

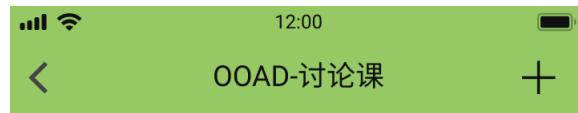
报名

```
GET /teacher/course/seminar/enrollList // 讨论课报名情况
data:
  klassSeminarID

model:{
  klassSeminarID:BigInteger
  seminarStatus:Int
  seminarName:String
  attendanceList:List<Attendance>
  attendance:Attendance
}

return:
  Page("student/course/seminar/enrollList")
```





第一组: 1-1 业务流程.ppt

第二组: 1-4 未提交

第三组: 可报名

第四组

第五组

第六组



( AJAX )

POST /teacher/course/seminar/enrollList/enroll //报名

data:

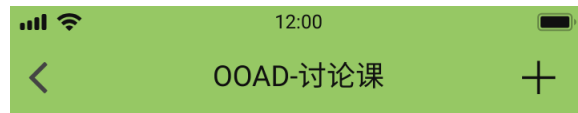
klassSeminarId

attendance

return:

报名成功:HTTPStatus(200)

报名失败:HTTPStatus(409)



第一组: 1-1

第二组: [可报名](#)

第三组: [可报名](#)

第四组: 1-3

第五组: [可报名](#)

第六组: 1-5

取消报名

( AJAX )

DELETE /teacher/course/seminar/enrollList/enroll

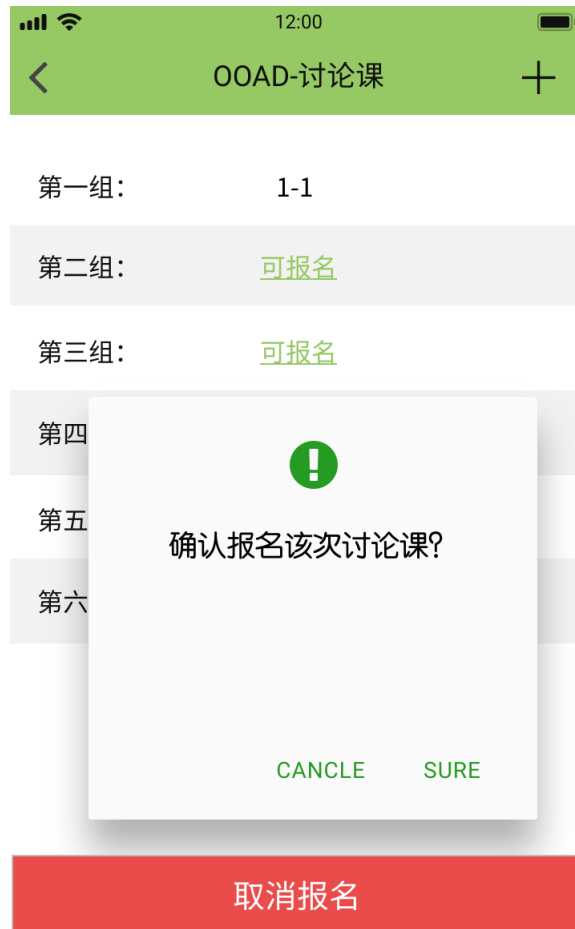
data:

klassSeminarId

return:

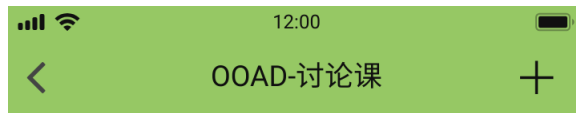
取消报名成功: HTTPStatus(200)

取消报名失败: HTTPStatus(409)



```
POST /teacher/course/seminar/enrollList/enroll
data:
  klassSeminarId
  attendance

return:
  报名成功:HTTPStatus(200 + "")
  报名失败:HTTPStatus(409 + "")
```



第一组: [1-1 业务流程.ppt](#)

第二组: [1-2 业务流程.ppt](#)

第三组: [1-4 业务流程.ppt](#)

第四组: [1-8 业务流程.ppt](#)

第五组: [1-7 业务流程.ppt](#)

第六组: [1-3 业务流程.ppt](#)

```
GET /teacher/course/seminar/PPT
```

```
data:
```

```
  klassSeminarId
```

```
  PPTurl
```

```
return:
```

```
  获取文件成功: HTTPStatus(200 + "")
```

```
  获取文件失败: HTTPStatus(404 + "")
```

12:00

OOAD-已报名

轮次：

第二轮

主题：

业务流程分析

课次序号：

第1次

要求

报名

课程

PPT:

PPT:

Choose File

No file chosen

CANCEL

SURE

进入讨论课

PPT提交

```
POST /teacher/course/seminar/PPT
data:
  klassSeminarId
  file (MutipartFile)

return:
  上传文件成功:HTTPStatus(200 + "")
  上传文件失败:HTTPStatus(409 + "")
```

12:00

OOAD-已报名

轮次:

第二轮

主题:

业务流程分析

课次序号:

第1次

要求

报名

课程

PPT:

书面:

PPT:

Choose File

No file chosen

CANCEL

SURE

书面报告提交

```
POST /teacher/course/seminar/PPT
data:
  klassSeminarId
  file (MutipartFile)

return:
  上传文件成功:HTTPStatus(200 + "")
  上传文件失败:HTTPStatus(409 + "")
```

12:00

<

OOAD-已报名

+

轮次:

第二轮

主题:

业务流程分析

课次序号:

第1次

要求:

界面导航图和所有界面原型设计  
课堂讨论每个小组15分钟

报名:

2016- (1) 第3组

课程情况:

已完成

报名情况

PPT:

已提交

书面报告:

已提交

查看成绩

GET /teacher/course/seminar/score // 讨论课成绩页

```
data:{
  klassSeminarID
  klassID
}

model:{
  //各项成绩
  //报名情况
  courseName:string
  score:Score
}
```

```
return:
  Page("student/course/seminar/grade")
```

轮次： 第二轮

主题： 业务流程分析

课次序号： 第1次

要求： 界面导航图和所有界面原型设计  
课堂讨论每个小组15分钟

报名： 2016-（1）第3组

课程情况： 正在进行 [报名情况](#)

PPT： 已提交

进入讨论课

PPT提交

```
GET /student/course/seminar/progressing
data:
  klassSeminarId
model:
  teamList:List<String>
  courseName:String
  seminarName:String
return:
  Page("teacher/course/seminar/progressing")
```



# 教师移动端 controller API

## 个人信息管理

- // 获取激活教师账户界面  
GET /teacher/activation  
return:  
Page("teacher/activation")
- // 激活时点击确认提交按钮  
(AJAX)  
POST /teacher/activation  
data:  
password  
email  
return:  
redirect:"/teacher/index"



The image shows a mobile application interface for setting a password. At the top, there is a status bar with signal strength, Wi-Fi, and the time 12:00. Below this is a navigation bar with a back arrow and the title "密码设置" (Password Setting). The main content area contains two input fields: "输入密码" (Enter Password) and "确认密码" (Confirm Password). Below these fields is a large, empty rectangular area. At the bottom, there is a green button labeled "确认提交" (Confirm Submit). A small text note at the bottom left states: "可包含数字、字母、下划线。长度不少于六位" (Can contain numbers, letters, and underscores. Length not less than six digits).

```
// 获取教师个人主页界面
GET /teacher/index
    if(账号未激活)
        redirect:"/teacher/activation"
    else
        model:
            addAttribute("curTeacher", Teacher)
    return:
        Page("TeacherIndex")
```



12:00



我



待办



个人页



讨论课



我的课程



账户设置



- ```
// 获取教师账户与设置页面
GET /teacher/setting
    model:
        addAttribute("curTeacher", Teacher)
        addAttribute("adminEmail", String)
    return:
        Page("setting")
```



12:00



## 账户与设置



姓名： 王老师

教工号： 333332222233333

联系方式（邮箱）： handsome@xmu.edu.cn

[修改](#)

账户密码



管理员邮箱： 654321@163.com

# 退出登录

- ```
// 修改邮箱按钮
GET /teacher/setting/modifyEmail
return:
Page("modifyEmailPage")
```
- ```
// 修改邮箱界面点击确认提交按钮
(AJAX)
POST /teacher/setting/modifyEmail
data:
  teacherNum
  email
return:
  @ResponseBody
  修改成功 : HTTPStatus(200)
  修改失败 : HTTPStatus(409)
```



The image shows a mobile application interface for modifying an account email. At the top, there is a status bar with signal strength, Wi-Fi, and the time 12:00. Below this is a navigation bar with a back arrow and the title '账户邮箱'. The main content area has a label 'e-mail:' followed by a text input field with the placeholder '填写新邮箱'. At the bottom, there is a green button labeled '确认提交'. A small note at the bottom left specifies the email format: '邮箱格式如: user@host.domainnames'.

```
// 获取账户密码修改页面
GET /teacher/setting/modifyPwd
return:
    Page("modifyPwdPage")
```

- // 修改密码点击确认提交按钮  
(AJAX)  
POST /teacher/setting/modifyPwd  
data:  
 teacherNum  
 password  
return:  
 @ResponseBody  
 修改成功: HTTPStatus(200)  
 修改失败: HTTPStatus(409)

- // 获取教师通知界面  
GET /teacher/notification  
return:  
 Page("teacher/notification")



12:00



## 待办及通知



J2EE—张老师的共享分组邀请



OOAD (1) 1111 王二 的特殊组队申请



OOAD 2016(1)李四 的不合法组队申请



### 历史消息:

OOAD 2016 (1) 小明 的特殊组队申请



OOAD 2016 (2) 大明 的特殊组队申请





## 课程管理

- ```
// 获取教师课程列表页
GET /teacher/course
model:
    addAttribute("courseList", List<Course>)
return:
    Page("teacher/courseList")
```



12:00



## 课程管理



OOAD (主)



学生成绩



学生组队



课程信息



班级信息



讨论课设置



共享设置



J2EE





## 新建课程



- ```
// 获取课程详情页
GET /teacher/course/info
data:
  courseId
model:
  addAttribute("curCourse", Course)
return:
  Page("courseInfo")
```



12:00



OOAD



---

## 课程简介

---

成绩计算规则：

课堂展示 %50

课堂提问 %10

书面报告 %30

小组人数：

6~8

组队开始时间：

2018.10.11 12:00

组队截止时间：

2018.11.11 12:00

组员性别要求：

男：2-4 女：2-4

组员星座要求：

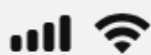
无

冲突课程：

.Net(XX老师)

## .Net(XX老师)

- ```
// 获取教师创建课程页
GET /course/create
return:
    Page("teacher/course/create")
```
- ```
// 创建课程行为
(AJAX)
PUT /teacher/course
data:
    course: CourseDTO
data-type:
    json
return:
    @ResponseBody
    创建成功: HttpStatus(200)
    因课程重名而失败: HttpStatus(409)
```



12:00



## 新建课程

课程名称

课程要求

成绩计算规则：

课堂展示

50%

课堂提问

20%

书面报告

30%

小组人数：

上限

下限

组队开始时间：

mm/dd/yyyy, --:-- --

组队截止时间：mm/dd/yyyy, --:-- --

组员性别要求：

男：上限 下限      女：上限 下限

是否根据星座分组：

冲突课程

.Net (\*\*老师) ▼

.Net (\*\*老师) ▼

新增对象

发布

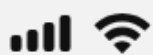
```
// 删除课程行为
( AJAX )
DELETE /teacher/course/{courseID}
return:
    @ResponseBody
    删除成功: HTTPStatus(200)
```

- ```
// 修改课程信息行为
( AJAX )
PATCH /teacher/course
data:
    course: CourseDTO
data-type:
    json
return:
    修改成功: HTTPStatus(200)
```

## 班级管理

- ```
// 获取班级列表页
POST /teacher/course/klassList
data:
    courseId
model:
    addAttribute("klassList", List<Klass>)
return:
    Page("teacher/course/klassList")
```





12:00



## 班级信息



### 2016 - 1

讨论课时间： 周三 7、8节

讨论课地点： 海韵教学楼

班级学生名单： 周三56节.xlsx

Choose File No fil...chosen

提交

删除班级

### 2016 - 2

讨论课时间： 周二 7、8节

讨论课地点： 海韵教学楼

班级学生名单： 当前无导入名单

Choose File No fil...chosen

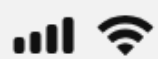
提交

删除班级



新增班级

- ```
// 获取教师创建班级页
GET /teacher/course/klass/create
model:
    addAttribute("courseId", courseId)
return:
    Page("teacher/course/klass/create")
```
- ```
// 创建班级行为
(AJAX)
PUT /teacher/course/klass
data:
    klass:KlassDTO
data-type:
    json
return:
    @ResponseBody
    创建成功: HTTPStatus(200 + authorities)
    因班级重名而创建失败: HTTPStatus(409)
```



12:00



## 新建班级

(2)

班级名：

年级

班级

上课时间：

上课地点：

班级学生名单：

Choose File

No fil...chosen

# 保存

- ```
//删除班级
( AJAX )
DELETE /course/class/{klassId}
return:
    @ResponseBody
    删除成功 : HttpStatus(200)
```

## 讨论课管理

- ```
// 讨论课列表页
GET /teacher/course/seminar
data:
    courseId
model:
    addAttribute("courseId", courseId)
    addAttribute("roundList", List<Round>)
    addAttribute("klassList", List<Klass>)
return:
    Page("teacher/course/seminarList")
```



//获取该轮轮次设置

POST /teacher/course/seminar/setting

data:

roundId

return:

Page("roundSetting")

- //修改轮次设置

PATCH /teacher/course/seminar/setting

data:

round:RoundDTO

return:

HTTPStatus(200)



12:00



## 第一轮

讨论课：

业务流程分析

领域模型设计

+ 添加讨论课

成绩设置：

展示：

最高分



提问：

平均分



报告：

最高分



本轮讨论课报名次数：

一班：

1（默认）



二班：

1（默认）



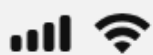
三班：

1（默认）



## 修改

- // 创建讨论课界面  
POST /teacher/course/seminar/create  
data:  
    courseId  
return:  
    Page("teacher/course/seminar/create")
- // 创建讨论课行为  
PUT /teacher/course/seminar/create  
data:  
    seminar:seminarDTO  
return:  
    新建成功 : HTTPStatus(200)



12:00



OOAD

## 新建讨论课

主题 讨论课大纲

讨论课主要内容 要求 具体讨论部分...

讨论课次序号:

1



讨论课可见:



可见

展示报名开始时间: mm/dd/yyyy, --:-- --

展示报名截止时间: mm/dd/yyyy, --:-- --

报名小组数:

6

报名顺序:



自定

所属Round:

无 (默认新建round)





# 发布

- ```
// 讨论课详情页
POST /teacher/course/seminar/info
data:
  klassId
  seminarID
model:
  addAttribute("klassSeminar", klassSeminar)
return:
  Page("teacher/course/seminar/info")
```



12:00



## OOAD-讨论课



轮次： 第二轮

主题： 领域模型

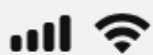
课次序号： 1

要求： 界面导航图和所有界面原型设计  
课堂讨论每个小组15分钟

课程情况： 未开始 [查看信息](#)

开始讨论课

- ```
// 讨论课修改
PATCH /teacher/course/seminar
  data:
    seminar:seminarDTO
  return:
    HTTPStatus(200)
```



12:00



OOAD

## 修改讨论课

### 业务流程分析

界面导航图和所有界面原型设计  
课堂讨论每个小组15分钟

讨论课次序号:

1



讨论课可见:



可见

展示报名开始时间:

mm/dd/yyyy, --:-- --

展示报名截止时间:

mm/dd/yyyy, --:-- --

报名小组数:

6

报名顺序:



自定

所属Round:

无（默认新建round）



书面报告提交截止时间:

2016-1

mm/dd/yyyy, --:-- --

2016-2

mm/dd/yyyy, --:-- --

删除该讨论课

保存

- ```
// 删除讨论课
DELETE /teacher/course/seminar/{seminarId}
return:
  删除成功 : HTTPStatus(200)
```

//讨论课报名列表

GET /teacher/course/seminar/enrollList

data:

klassSeminarId

model:

addAttribute("enrollList", List<Attendance>)

return:

Page("teacher/course/seminar/enrollList")



12:00



## OOAD-讨论课



第一组： 1-1 业务流程.ppt

第二组： 1-2 未提交

第三组： 未报名

第四组： 1-3 业务流程.ppt

第五组： 1-4 未提交

第六组： 1-5 业务流程.ppt

- //获取讨论课正在进行页面  
(websocket)  
GET /teacher/course/seminar/progressing  
data:  
    klassSeminarId  
model:  
    addAttribution("enrollList", List<Attendance>)  
return:  
    Page("teacher/course/seminar/progressing")





12:00



## OOAD-讨论课



### 业务流程分析



1-3 组展示： 5 : 00 被提问： 1 : 00 已有1位同学提问

1-5

1-2

**1-3**

1-4

1-6

1-8

### 展示分数

成绩输入

抽取提问

## 下组展示

- `//已经结束的讨论课书面报告页面`  
`GET /teacher/course/seminar/finished`  
`data:`  
`klassSeminarId`  
`model:`  
`addAttribution("enrollList", List<Attendance>)`  
`return:`  
`Page("teacher/course/seminar/finished")`



12:00



## OOAD-书面报告



第一组: [1-1 业务流程.dox](#)

5.0

第二组: [1-3 业务流程.dox](#)

0.0

第三组: [1-2 业务流程.dox](#)

0.0

第四组: [1-7 业务流程.dox](#)

成绩

第五组: [1-5 业务流程.dox](#)

4.0

第六组: [1-4 业务流程.dox](#)

3.0

批量下载

确认

- `//讨论课分数详情页`  
`GET /teacher/course/seminar/grade`  
`data:`  
`klassSeminarId`  
`model:`  
`addAttribute("score",SeminarScore)`  
`return:`  
`Page("teacher/course/seminar/grade")`



12:00



## OOAD-书面报告



展示      提问      书面报告      总分

1-1:      5.0      5.0      5.0      5.0

1-3:      5.0      5.0      5.0      5.0

1-2:      5.0      5.0      5.0      5.0

1-7:      5.0      5.0      5.0      5.0

1-5:      5.0      5.0      5.0      5.0

1-4:      5.0      5.0      5.0      5.0

确认

修改

## 组队管理

- ```
// 课程下获取队伍列表页
POST /teacher/course/teamList
data:
  courseId
model:
  addAttribute("teamList", List<Team>)
return:
  Page("teacher/course/teamList")
```



12:00



# 学生组队



1-1

HHHH

invaild



组长:

1111111111111111

王二

组员:

1111111111111111

~~~

1111111111111111

~~~

1111111111111111

~~~

1111111111111111

~~~

2-1

AAAA



3-1

BBBB



- `//课程下获取学生成绩页`  
`GET /teacher/course/grade`  
    `data:`  
        `courseId`  
    `model:`  
        `addAttribute("roundScore", RoundScore)`  
    `return:`  
        `Page("teacher/course/grade")`





12:00



## 学生成绩



### 第一轮



1-1

4.50分



1-2

5.00分



1-3

4.75分



2-1

4.50分



### 用例分析

展示: 5.0

提问: 5.0

书面报告: 4.50

### 界面原型

展示: 5.0

提问: 5.0

书面报告: 4.50

### 总成绩

展示: 5.0

提问: 5.0

书面报告: 4.50

修改成绩

第二轮



导出成绩