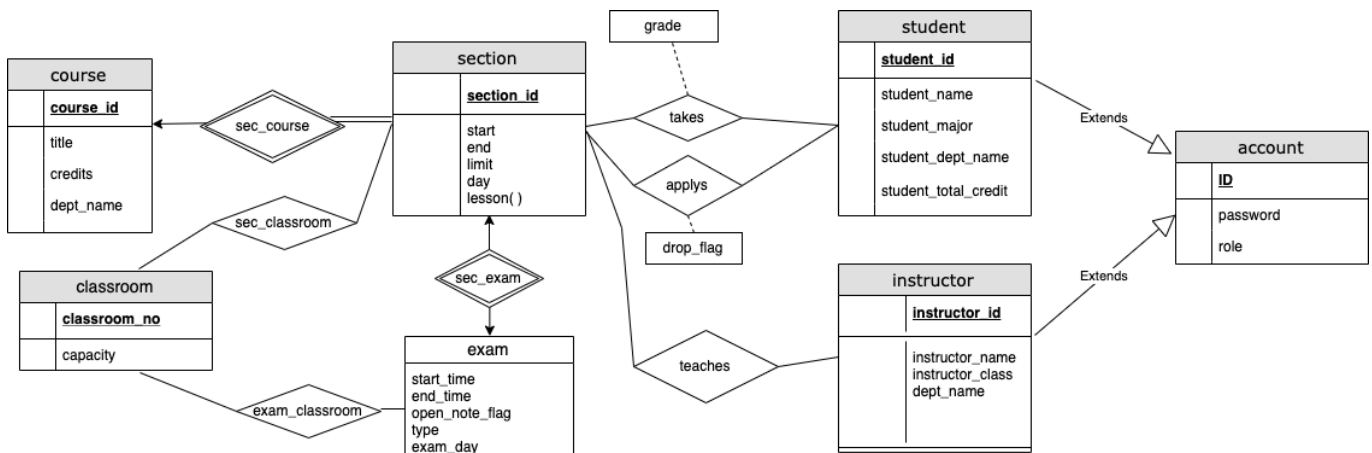


项目文档（《数据库设计》课程）

选课系统数据库设计



从ER图到表结构

实体集

- 课程信息 course (course_id, title, credits, dept_name)
- 开课信息 section (course_id, section_id, start, end, classroom_no, limit, day, lesson)
- 账户信息 account (ID, password, role)
- 学生信息 student(student_id, student_name, student_major, student_dept_name, student_total_credit)
- 教师信息 instructor (instructor_id, instructor_name, instructor_class, dept_name)
- 考试信息 exam (course_id, section_id, exam_classroom_no, exam_day, type, start_time, end_time, open_note_flag)
- 教室信息 classroom (classroom_no, capacity)

关系集

- 教师 & 开课：教授 teaches (instructor_id, course_id, section_id)
- 学生 & 开课：选课 takes (course_id, section_id, student_id, grade)
- 学生 & 开课：申请 application (course_id, section_id, student_id, status, application_reason, if_drop)

表结构分析

- course: 用于存储和管理课程信息的数据库表，主键是课程编号course_id。

```
CREATE TABLE "course" (
  "course_id" TEXT(255) NOT NULL,
  "title" TEXT(255) NOT NULL,
  "credits" integer(10) NOT NULL,
  "dept_name" TEXT(255),
```

```
PRIMARY KEY ("course_id")
);
```

- **section**: 用于存储和管理本学期实际上开设的课程信息，是一个弱实体集。开课涉及到上课时间，如星期五 3-4这样的时间。为了保证原子性，系统拆分成了三部分：day表示是一周的第几天，start表示这门课的开始节次，end表示这门课的结束节次。lesson作为导出属性而存在。section 作为弱实体集，它的主键由course_id和section_id联合组成。本数据库表引用了两个外键，一个是course的主键，一个是classroom的主键。针对这两个外键，数据库库表都设置了级联删除以及级联更新。

```
CREATE TABLE "section" (
  "course_id" TEXT(255) NOT NULL,
  "section_id" INTEGER(10) NOT NULL,
  "classroom_no" TEXT(255),
  "limit" INTEGER(10) NOT NULL,
  "day" INTEGER(10) NOT NULL,
  "start" integer(10) NOT NULL,
  "end" integer(10) NOT NULL,
  PRIMARY KEY ("course_id", "section_id"),
  FOREIGN KEY ("course_id") REFERENCES "course" ("course_id") ON DELETE
  CASCADE ON UPDATE CASCADE,
  FOREIGN KEY ("classroom_no") REFERENCES "classroom" ("classroom_no") ON
  DELETE CASCADE ON UPDATE CASCADE
);
```

- **account**: 用于存储管理账户信息的数据库表，含有用户名，密码和角色。主键是用户名ID。

```
CREATE TABLE "account" (
  "ID" text(255) NOT NULL,
  "password" text(255) NOT NULL,
  "role" integer(10) NOT NULL,
  PRIMARY KEY ("ID")
);
```

- **student**: 用于存储和管理学生的相关信息。信息主要包括学号，姓名，专业，院系和总学分。主键是学号student_id。总学分的默认值是0。

```
CREATE TABLE "student" (
  "student_id" TEXT(255) NOT NULL,
  "student_name" TEXT(255) NOT NULL,
  "student_major" TEXT(255) NOT NULL,
  "student_dept_name" TEXT(255) NOT NULL,
  "student_total_credit" INTEGER(10) DEFAULT 0,
  PRIMARY KEY ("student_id")
);
```

- instructor: 用于存储和管理教师的相关信息。信息主要包括教师编号, 教师姓名, 所属院系和职称。主键是教师编号。

```
CREATE TABLE "instructor" (  
  "instructor_id" TEXT(255) NOT NULL,  
  "instructor_name" TEXT(255),  
  "instructor_class" TEXT(255),  
  "dept_name" TEXT(255),  
  PRIMARY KEY ("instructor_id")  
);
```

- exam: 用于存储和管理课程考核的相关信息。含有属性课程编号, 开课编号, 考试类型(论文或者考试)等。其中教室, 考试的开始时间以及是否开卷是在考核类型为考试时才有涵义。考核作为一个弱实体集, 它的主键由课程编号和开课编号联合组成, 与开课一一对应。本表引用了三个外键, 分别是表section的联合主键和表classroom的主键。删除和更新都是级联执行。

```
CREATE TABLE "exam" (  
  "course_id" TEXT(255) NOT NULL,  
  "section_id" TEXT(255) NOT NULL,  
  "exam_classroom_no" TEXT(255),  
  "exam_day" integer(5) NOT NULL,  
  "type" integer(5) NOT NULL,  
  "start_time" TEXT(255),  
  "end_time" TEXT(255) NOT NULL,  
  "open_note_flag" integer(5),  
  PRIMARY KEY ("course_id", "section_id"),  
  FOREIGN KEY ("course_id", "section_id") REFERENCES "section"  
  ("course_id", "section_id") ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY ("exam_classroom_no") REFERENCES "classroom"  
  ("classroom_no") ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- classroom: 教室信息。含有教室的编号以及容量。主键是教室编号。
- teaches: 教师教授课程关系集对应的数据库表。主键是course_id, section_id和instructor_id。相关外键的删除和更新都是级联的。

```
CREATE TABLE "teaches" (  
  "instructor_id" TEXT(255) NOT NULL,  
  "course_id" TEXT(255) NOT NULL,  
  "section_id" INTEGER(10) NOT NULL,  
  PRIMARY KEY ("instructor_id", "course_id", "section_id"),  
  FOREIGN KEY ("course_id", "section_id") REFERENCES "section"  
  ("course_id", "section_id") ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY ("instructor_id") REFERENCES "instructor" ("instructor_id")  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- takes: 学生选课关系集对应的数据库表。主键是course_id, section_id和student_id。相关外键的删除和更新都是级联的。

```
CREATE TABLE "takes" (  
  "course_id" text(255) NOT NULL,  
  "section_id" INTEGER(10) NOT NULL,  
  "student_id" text(255) NOT NULL,  
  "grade" TEXT(10),  
  PRIMARY KEY ("course_id", "section_id", "student_id"),  
  FOREIGN KEY ("course_id", "section_id") REFERENCES "section"  
  ("course_id", "section_id") ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY ("student_id") REFERENCES "student" ("student_id") ON DELETE  
  CASCADE ON UPDATE CASCADE  
);
```

- application: 学生选课申请对应的数据库表。主键是course_id, section_id和student_id。相关外键的删除和更新都是级联的。特别的含有属性if_drop用来标志学生是否申请成功了改课程之后又退掉了。相关外键的删除和更新都是级联的。

```
CREATE TABLE "application" (  
  "course_id" text(255) NOT NULL,  
  "section_id" integer(5) NOT NULL,  
  "student_id" text(255) NOT NULL,  
  "status" integer(5) NOT NULL DEFAULT 0,  
  "application_reason" text(255),  
  "if_drop" integer(5) DEFAULT 0,  
  PRIMARY KEY ("course_id", "section_id", "student_id"),  
  FOREIGN KEY ("student_id") REFERENCES "student" ("student_id") ON DELETE  
  CASCADE ON UPDATE NO ACTION,  
  FOREIGN KEY ("course_id", "section_id") REFERENCES "section"  
  ("course_id", "section_id") ON DELETE CASCADE ON UPDATE NO ACTION  
);
```

函数依赖与范式分析

- 所有关系的属性的域都是原子的，不包含任何组合属性，所以上述关系模式都属于第一范式。
- 没有任何非主属性部分依赖于键，所以也符合第二范式。
- 没有任何非主属性传递依赖于键，所以符合第三范式。
- 没有任何属性传递依赖于键，所以符合BC范式。

综上，我们设计的关系模式属于BC范式。

功能点实现——数据库操作逻辑

选课

选退课功能开放后，学生才可以进行选课。学生选课，数据库需要先检查该课程的选课人数是否已经达到选课上限，并且检查选择的课程是否与已选课程具有时空冲突（包括上课和考试的时空冲突）。如果没超过上限并且没有冲突，则该学生可以选上该门课程。

退课

选退课功能开放后，学生才可以进行退课。学生退课，如果学生已经选择这门课程，直接去掉选课表中的一行数据。注意，如果用户是通过选课申请选上该门课程，需要标记选课申请记录的退课标志使得该生不能再申请该门课程。

选课申请

选退课功能开放后，学生才可以填写课程申请。当选课人数达到上限时，学生才能进行选课申请。当教室容量不允许的人上课时或者该学生已经申请成功过这门课程但是退掉了的时候，系统会直接决绝掉他的选课申请。

分数导入

登分系统开放后，任课老师需要将学生的成绩导入系统中。在分数登录的界面中，系统提供了分数的样例文件。教师课进行下载并填入相应条目。对于每条课程成绩，系统会先检查成绩中学生是否选过这门课程。没有选择过这门课程，系统会给出提示并拒绝导入这门成绩，但错误之前的成绩都会成功导入。如果成绩重复导入以后导入的为准。

课程申请处理

学生进行选课申请，任课老师可以选择同意或者拒绝。如果同意，则会在takes数据库表中加入记录，并更新申请记录。如果拒绝，则只会更新选课申请记录未已拒绝。

学生导入

导入学生时，系统会先检查学生的编号是否在student表中已经存在，即是否导入已经存在的学生的信息。如果编号不存在，并且其他信息完整，则支持本次导入。

老师导入

导入老师时，系统依旧只是检查相应的编号时否在instructor表中已经存在。如果不存在且其他信息填写完整，则可以导入。

课程信息导入

导入课程时，系统依旧只是检查相应的编号时否在course表中已经存在。如果不存在且其他信息填写完整规范，则可以导入。

开课信息导入

开课信息的导入需要检查的信息繁多。首先需要在section表中检查该开课是否已经存在，同时需要检查course是否存在和老师是否存在。如果课程没有被重复开设，并且相应的course和老师都存在，则检查课程在老师和教室上是否存在时空冲突。如果有冲突，禁止导入并给出提示。

考试信息导入

考试信息的导入是建立在课程已经开始的情况下，如果课程还没由开设当然需要拒绝导入。除此以外，需要检查对于同一门开课是否已经导入过考试信息。重复导入不被允许，系统提供过修改的接口。此外，系统还要检查考试是否具有时空冲突。对于考核方式为论文的考核，我们认为不存在任何的冲突。而对于考试，就需要仔细检查。

选课系统后端架构——Django Sqlite Python3

Django是一个开放源代码的Web应用框架，由Python写成。采用了MTV的框架模式，即模型M，视图V和模板T。

选课系统后端API说明

说明：所有调用的成功状态由返回的 HTTP 状态码表示，200 代表调用成功，返回的内容为各 API 自己的返回值，400 代表调用失败，返回以下格式的 JSON：

```
{
  "code": -1,
  "message": ""
}
```

OR

```
{
  "code": 1,
  "message": ""
}
```

以下为通用message：

message	含义	可能出现该message的 API
server error	服务器内部错误	全部
unauthorized	当前用户没有权限执行该操作或session已过期	需要特定用户权限的 API

对于 GET 请求，参数放在 URL 中；对于 POST 请求，参数 放在 body 中。

所有 API 的参数若非特殊说明均为必选参数。

登录

POST /selectCourse/login/

参数列表

名称	类型	描述
user_id	string	学号或者工号

名称	类型	描述
password	string	密码

返回值

```
{
  "user_name": "黄佳妮",
  "role": 1
}
```

message

message	含义
wrong userid	用户名不存在
wrong password	密码错误
login successfully	登陆成功

登出

/selectCourse/logout/

参数列表

不需要参数

返回值

无返回值

message

message	含义
logout successfully	登出成功

选课

POST /selectCourse/select/

参数列表

名称	类型	描述
user_id	string	学号
course_id	string	学校课程唯一代码

名称	类型	描述
section_id	string	课程类别唯一代码

返回值

无

message

message	含义
wrong course id	课程代码错误
wrong section id	课程类别代码错误
select successfully	选课成功
course with no vacancy	课程无余量
already selected	已经选过该课程
section time conflict	课程时间冲突
exam time conflict	考试时间冲突

退课

POST /selectCourse/drop/

参数列表

名称	类型	描述
user_id	string	学号
course_id	string	学校课程唯一代码
section_id	string	课程类别唯一代码

返回值

无

message

message	含义
drop error: haven't taken yet	未选修该课程
drop successfully	退课成功

查询已选课列表

POST /selectCourse/checkCourseTable

参数列表

名称	类型	描述
user_id	string	用户 ID（留空则默认获取当前已登录用户的信息）
current_page_num	int	用户 ID（留空则默认获取当前已登录用户的信息）

返回值

```
{
  "total_num" : 1,
  "sections" : {
    "title": "大气环境科学",
    "course_id": "ATM000000003",
    "section_id":1,
    "dept_name":"航空航天系",
    "instructor_name":"curry",
    "credits":2,
    "classroom_no":"Z2202",
    "day":2,
    "start": 3,
    "end": 4
  }
}
```

message

message	含义
show course table	显示课程信息

查询所有课程信息(学生和root权限)

POST /selectCourse/checkAllCourses

参数列表

名称	类型	描述
user_id	string	用户 ID（留空则默认获取当前已登录用户的信息）
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "sections" : {
    "title": "大气环境科学",
    "course_id": "ATM000000003",
    "section_id":1,
    "dept_name":"航空航天系",
    "credits":2,
    "classroom_no":"Z2202",
    "day":2,
    "start": 3,
    "end": 4
  }
}
```

message

无

显示学生/教师基本信息

POST /selectCourse/checkPersonalInfo

参数列表

名称	类型	描述
user_id	string	用户 ID

返回值

对于学生

```
{
  "student_id": "17302010063",
  "student_name": "黄佳妮",
  "student_major": "软件工程",
  "student_dept_name":"软件学院",
  "student_total_creidt":95,
  "student_gpa":4.0,
  "student_grade":[
    {
      "course_id": "MATH1000001",
      "section_id": 1,
      "grade": "A"
    }
  ]
}
```

```
}
```

对于教师

```
{
  "instructor_id": "SOFT00000001",
  "instructor_name": "王小明",
  "instructor_class": "副教授",
  "dept_name": "软件学院"
}
```

message

message	含义
show personal info	显示用户信息

搜索课程

GET /selectCourse/search

参数列表

名称	类型	描述
user_id	string	用户 ID
course_id	string	根据course_id和section_id进行搜索
section_id	int	根据course_id和section_id进行搜索
title	string	根据课程名称进行搜索
instructor_name	string	根据教师名字进行搜索
dept_name	string	根据开课院系进行搜索

返回值

```
{
  "total_num" : 1,
  "sections" : {
    "title": "大气环境科学",
    "course_id": "ATM000000003",
    "section_id": 1,
    "dept_name": "航空航天系",
    "credits": 2,
  }
}
```

```
    "classroom_no": "Z2202",  
    "day": 2,  
    "start": 3,  
    "end": 4  
  }  
}
```

message

message	含义
search suceessfully	搜索成功
no search result	无搜索结果

提交选课申请

POST /selectCourse/submitApplication

参数列表

名称	类型	描述
user_id	string	
course_id	string	
section_id	int	
application_reason	string	申请理由

返回值

无

message

message	含义
can't apply course which is already applied 不能申请已申请课程 can't apply selected course 不能申请已选课程 can't apply dropped course 不能申请已退的申请通过课程 apply successfully 提交成功 can't apply course with vacancy 不能申请有余量课程 exceed the classroom capacity 不能申请人数已超过教室容量的课程	

处理选课申请

POST /selectCourse/handleApplication

参数列表

名称	类型	描述
----	----	----

名称	类型	描述
user_id	string	
course_id	string	
section_id	int	
status	int	处理状态

返回值

无

message

message	含义
handle successfully	处理成功

查看选课申请

学生和老*师*都可查看其对应的选课申请

POST /selectCourse/handleApplication

参数列表

名称	类型	描述
user_id	string	

返回值

```
{
  "total_num" : 1,
  "applications" : {
    "course_id": "ATM000000003",
    "section_id": 1,
    "student_id": "17302010063",
    "status": 1,
    "application_reason": "I love this course",
    "if_drop": 1
  }
}
```

message

message	含义
check application info	显示选课申请信息

查看所教授课程（教师权限）

POST /selectCourse/checkTaughtCourses/

参数列表

名称	类型	描述
user_id	string	

返回值

```
{
  "total_num" : 1,
  "sections" : [
    {
      "title": "大气环境科学",
      "course_id": "ATM000000003",
      "section_id":1,
      "dept_name":"航空航天系",
      "credits":2,
      "classroom_no":"Z2202",
      "day":2,
      "start": 3,
      "end": 4
    }
  ]
}
```

message

message	含义
show course table	显示课程信息

查看该课程花名册（教师权限）

POST /selectCourse/checkCourseNameList/

参数列表

名称	类型	描述
user_id	string	

返回值

```
{
  "total_num" : 1,
```

```
    "sections" : [
      {
        "title": "大气环境科学",
        "student_id": "17302010063",
        "student_name": "黄佳妮",
        "student_major": "航空航天系",
        "student_dept_name": "航空航天大学",
        "grade": "A"
      }
    ]
  }
```

message

message	含义
show course table	显示课程信息

查看课程考试列表（学生权限）

POST /selectCourse/checkExamTable/

参数列表

名称	类型	描述
user_id	string	

返回值

```
{
  "total_num" : 1,
  "sections" : [
    {
      "title": "大气环境科学",
      "student_id": "17302010063",
      "student_name": "黄佳妮",
      "student_major": "航空航天系",
      "student_dept_name": "航空航天大学",
      "grade": "A"
    }
  ]
}
```

message

message	含义
show course table	显示课程信息

自动导入模块

下载模版文件（老师）

GET /selectCourse/downloadFile

参数列表

名称	类型	描述
file_type	int	STUDENT_FILE = 1 COURSE_FILE = 2 SCORE_FILE = 3 SECTION_FILE = 4 INSTRUCTOR_FILE = 5

导入学生课程成绩（教师权限）

导入课程信息（root权限）

导入学生信息（root权限）

导入教师信息（root权限）

导入考试信息（root权限）

管理员增删改查统一接口说明

增 Insert

手动导入课程信息

POST /selectCourse/insertCourse/

参数列表

名称	类型
course_id	string
title	string
credits	int
dept_name	string

返回值

None

message

message	含义
---------	----

message	含义
insert error: already exist	课程已存在
handle successfully ok	

手动导入开课信息

POST /selectCourse/insertSection/

参数列表

名称	类型
course_id	string
section_id	string
time	string
classroom_no	int
lesson	int
limit	int
day	int

返回值

None

message

message	含义
insert error: already exist	开课已存在
handle successfully ok	

手动导入学生信息

POST /selectCourse/insertStudent/

参数列表

名称	类型
student_id	string
student_name	string
student_major	string

名称	类型
student_dept_name	string
student_total_credit	string

返回值

None

message

message	含义
insert error: already exist	学生已存在
handle successfully ok	

手动导入教师信息

POST /selectCourse/insertInstructor/

参数列表

名称	类型
instructor_id	string
instructor_name	string
instructor_class	string
dept_name	string

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

手动导入教室信息

POST /selectCourse/insertClassroom/

参数列表

名称	类型
classroom_no	string
capacity	int

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

手动导入账户信息

POST /selectCourse/insertAccount/

参数列表

名称	类型
user_id	string
password	string
role	int

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

手动导入考试信息

POST /selectCourse/insertExam/

参数列表

名称	类型
course_id	string
title	string
credits	int
dept_name	string

返回值

None

message

message	含义
insert error: already exist	考试已存在
handle successfully ok	

删

删除课程信息

POST /selectCourse/deleteCourse/

参数列表

名称	类型
course_id	string

返回值

None

message

message	含义
delete nonexist course	课程已存在
handle successfully ok	

删除开课信息

POST /selectCourse/deleteSection/

参数列表

名称	类型
course_id	string
section_id	string

返回值

None

message

message	含义
delete nonexistent section	课程已存在
handle successfully ok	

删除学生信息

POST /selectCourse/deleteStudent/

参数列表

名称	类型
student_id	string

返回值

None

message

message	含义
delete nonexistent student	课程已存在
handle successfully ok	

删除教师信息

POST /selectCourse/deleteInstructor/

参数列表

名称	类型
instructor_id	string

返回值

None

message

message	含义
delete nonexistent instructor	课程已存在
handle successfully ok	

删除教室信息

POST /selectCourse/deleteClassroom/

参数列表

名称	类型
classroom_no	string

返回值

None

message

message	含义
delete nonexistent classroom	课程已存在
handle successfully ok	

删除账户信息

POST /selectCourse/deleteAccount/

参数列表

名称	类型
user_id	string

返回值

None

message

message	含义
delete nonexistent account	课程已存在
handle successfully ok	

删除考试信息

POST /selectCourse/deleteExam/

参数列表

名称	类型
course_id	string
section_id	int

返回值

None

message

message	含义
delete nonexistent exam	课程不存在
handle successfully ok	

改

修改课程信息

POST /selectCourse/updateCourse/

参数列表

名称	类型
course_id	string
title	string
credits	int
dept_name	string

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

修改开课信息

POST /selectCourse/updateSection/

参数列表

名称	类型
course_id	string
section_id	string
time	string
classroom_no	int
lesson	int
limit	int
day	int

返回值

None

message

message	含义
insert error: already exist	开课已存在
handle successfully ok	

修改学生信息

POST /selectCourse/updateStudent/

参数列表

名称	类型
student_id	string

名称	类型
student_name	string
student_major	string
student_dept_name	string
student_total_credit	string

返回值

None

message

message	含义
insert error: already exist	学生已存在
handle successfully	ok

修改教师信息

POST /selectCourse/updateInstructor/

参数列表

名称	类型
instructor_id	string
instructor_name	string
instructor_class	string
dept_name	string

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully	ok

修改教室信息

POST /selectCourse/updateClassroom/

参数列表

名称	类型
classroom_no	string
capacity	int

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

修改账户信息

POST /selectCourse/updateAccount/

参数列表

名称	类型
user_id	string
password	string
role	int

返回值

None

message

message	含义
insert error: already exist	课程已存在
handle successfully ok	

修改考试信息

POST /selectCourse/updateExam/

参数列表

名称	类型
course_id	string
section_id	string
classroom_no	int
day	int
type	int
start_time	string
end_time	string
open_note_flag	int

返回值

None

message

message	含义
update error: nonexistent	课程不存在
handle successfully ok	

查

查看课程信息

POST /selectCourse/checkCourses/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "courses" : [
    {
      "title": "大气环境科学",
```

```
        "course_id": "ATM000000003",
        "dept_name": "航空航天系",
        "credits": 2
      }
    ]
  }
```

message

无

查看开课信息

POST /selectCourse/checkSections/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "sections" : {
    "course_id": "ATM000000003",
    "section_id": 1,
    "credits": 2,
    "classroom_no": "Z2202",
    "day": 2,
    "start": 3,
    "end": 4
  }
}
```

message

无

查看学生信息

POST /selectCourse/checkStudents/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "students" : {
    "student_id": "17302010063",
    "student_name": "黄佳妮",
    "student_major": "软件工程",
    "student_dept_name": "软件学院",
    "student_total_creidt": 95
  }
}
```

message

无

查看教师信息

POST /selectCourse/checkInstructors/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "instructors" : {
    "instructor_id": "SOFT00000001",
    "instructor_name": "王小明",
    "instructor_class": "副教授",
    "dept_name": "软件学院"
  }
}
```

message

无

查看教室信息

POST /selectCourse/checkClassrooms/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "classrooms" : {
    "classroom_no": "Z2202",
    "capacity": 90
  }
}
```

message

无

查看账户信息

POST /selectCourse/checkAccounts/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{
  "total_num" : 1,
  "ret_num": 15,
  "accounts" : [{
    "id": "www",
    "password": "www",
    "role": 1
  }
]
```

```
    } ]  
  }
```

message

无

查看考试信息

POST /selectCourse/checkExams/

参数列表

名称	类型	描述
current_page_num	int	当前页数

返回值

```
{  
  "total_num" : 1,  
  "ret_num": 15,  
  "exams" : [{  
    "course_id": "ATM0130004",  
    "section_id":1,  
    "classroom_no":"Z2204",  
    "day":5,  
    "type":1,  
    "start_time":"13:00",  
    "end_time":"15:00",  
    "open_note_flag":1  
  } ]  
}
```

message

无