

数据库课程设计接口定义

零、示例接口

- 请求方法: GET
- 请求url: /api/v1/paint-egg/
- 请求功能描述: 只是一个彩蛋接口罢了, 亲们可以在上线后试一下postman或curl调它

请求方法、功能描述由前端写, url由后端写

- 入参传递方式: 无(可选Query(URL、FormData(Body)、Json(Body))
- 入参定义: 无(统一使用Json写法定义参数, 下为例子)

```
1 //JSON
2 {
3     "字段名称": "字段值", // json 支持float、string、bool、array、object、null等类型
4     "数字": 0.0, //json没有整数, 全是浮点数
5     "字符串": "abcd",
6     "布尔" : true,
7     "数组" : [
8         1,2,3,4,5,6
9     ],
10    "对象" : {
11        "字段名称": "字段值",
12        "数字": 0.0,
13        "字符串": "abcd",
14        "布尔" : true,
15        "数组" : [
16            1,2,3,4,5,6
17        ]
18    },
19    "空": null //最好别有这个东西
20 }
```

如果你想要写一些json不支持, 但是别的格式支持的类型, 请用字符串占位, 注释来说明类型。

- 返回情况一
 - 返回码: 200
 - 返回体: response统一采用json返回, 固定格式如下

```

1 {
2     "code" : 0, //状态标记码
3     "data" : {
4         //自定义返回的数据格式，这里可以按照上述的JSON规则编写
5     },
6     "msg" : "响应消息"
7 }

```

如果有多个返回可能则需要写多种情况，如参数错误（400）、服务内部错误(500)等

一、正式接口由这里开始书写

User:

1. 登录

- 请求方法: POST
- 请求url: /api/v1/auth/login
- 请求功能描述：用户登录
- 入参传递方式：json
- 入参定义:

```

1 {
2     "userid": xxxxx, //公钥加密后的id
3     "password": "xxxxx", //公钥加密后的密码
4     "usertype": 1/2/3, // student/teacher/admin
5 }

```

- 返回情况一：登录成功
 - 返回码：200
 - 返回体:

```

1 {
2     "code" : 0,
3     "data" : null,
4     "msg" : "登录成功"
5 }

```

- 返回情况二：密码错误

- 返回码：401
- 返回体:

```
1 {  
2     "code" : -1,  
3     "data" : null,  
4     "msg" : "密码错误"  
5 }
```

- 返回情况三：无此用户或用户未激活

- 返回码：404
- 返回体:

```
1 {  
2     "code" : -2,  
3     "data" : null,  
4     "msg" : "无此用户或用户未激活"  
5 }
```

- 返回情况四：请求参数非法

- 返回码：400
- 返回体:

```
1 {  
2     "code" : -3,  
3     "data" : null,  
4     "msg" : "请求信息非法"  
5 }
```

- 返回情况四：服务端错误

- 返回码：500
- 返回体:

```
1 {  
2     "code" : 500,
```

```
3     "data" : null,
4     "msg" : "错误信息"
5 }
```

2. 注册

- 请求方法: POST
- 请求url: /api/v1/auth/register
- 请求功能描述: 用户注册
- 入参传递方式: json
- 入参定义:

```
1 {
2     "userid": "xxxxx",    //公钥加密后的id
3     "email": "xxxxxxx"
4     "password": "xxxxx"  //公钥加密后的密码
5 }
```

- 返回情况一: 登录成功
 - 返回码: 200
 - 返回体:

```
1 {
2     "code" : 0,
3     "data" : null,
4     "msg" : "注册成功"
5 }
```

- 返回情况二: 重复注册
 - 返回码: 400
 - 返回体:

```
1 {
2     "code" : -1,
3     "data" : null,
```

```
4     "msg" : "用户已注册"
5 }
```

- 返回情况三：无此用户id

- 返回码：404
- 返回体：

```
1 {
2     "code" : -3,
3     "data" : null,
4     "msg" : "无此用户id"
5 }
```

- 返回情况四：请求参数非法

- 返回码：400
- 返回体：

```
1 {
2     "code" : -3,
3     "data" : null,
4     "msg" : "请求信息非法"
5 }
```

- 返回情况四：服务端错误

- 返回码：500
- 返回体：

```
1 {
2     "code" : 500,
3     "data" : null,
4     "msg" : "错误信息"
5 }
```

3. 登出

- 请求方法: POST
-

- 请求url: /api/v1/auth/logout
- 请求功能描述: 用户注销
- 入参传递方式: 无
- 入参定义: 无
- 返回情况一: 登出成功
 - 返回码: 200
 - 返回体:

```
1 {  
2     "code" : 0,  
3     "data" : null,  
4     "msg" : "登出成功"  
5 }
```

- 返回情况二: 用户未登录
 - 返回码: 404
 - 返回体:

```
1 {  
2     "code" : -1,  
3     "data" : null,  
4     "msg" : "用户未登录"  
5 }
```

- 返回情况三: 状态更新失败
 - 返回码: 500
 - 返回体:

```
1 {  
2     "code" : -2,  
3     "data" : null,  
4     "msg" : "状态更新失败"  
5 }
```

4. 获取登录状态

- 请求方法: GET
- 请求url: /api/v1/auth/status
- 请求功能描述: 获取当前用户的登录状态
- 入参传递方式: json
- 入参定义: 无入参
- 返回数据格式:

```
1 {
2     "code":0,
3     "msg":"success",
4     "data":{
5         "loggedIn" : true,           // 是否成功登录
6         "userid": "",
7         "username": "",
8         "usertype" : 1/2/3,        // student/teacher/admin
9         "permission": 0
10    }
11 }
```

Student:

4.1 select

- 请求方法: POST
- 请求url: /api/v1/student/course/select
- 请求功能描述: 学生选择课程id为courseId的课程。
- 入参传递方式: json
- 无返回值
- 入参定义:

```
1 {
2     "courseId": 21,
3 }
```

4.2 getPageCount

- 请求方法: GET
- 请求url: /api/v1/student/course/page/count
- 请求功能描述: 根据课程名和教师名, 返回筛选过的课程页数 (总的符合条件的课程数 / 每个页面最多可显示的课程数【可手动设置: 比如这里暂定为'6'】)。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "courseName": "xxxxx",  
3     "teacherName": "xxxxxx" ,  
4     "pageSize": 6  
5 }
```

- 返回数据格式:

```
1 {  
2     "code":0,  
3     "msg":"success",  
4     "data":{  
5         "count": 12,  
6     }  
7 }
```

4.3 getPage

- 请求方法: GET
- 请求url: /api/v1/student/course/page/:index
- 请求功能描述: 根据课程名+教师名+访问页index, 返回筛选过的所有课程。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "courseName": "xxxxx",  
3     "teacherName": "xxxxxx",
```



```
4     "pageSize": 6,  
5 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4     "data": [  
5         {  
6             "courseId": 1,  
7             "courseName": "Math",  
8             "teacherName": "一个老师",  
9             "credit" : 2,  
10            "time": "3-3-2" // 星期几-第几节开始-上几节课  
11            "selectedCount": 2, // 已选课人数  
12            "maxSize": 20, // 课程最大容量  
13            "departmentName": "xx系", // 系名  
14            "grade": 2020 // 哪个年级上的课  
15        },  
16        .....  
17    ]  
18 }
```

4.4 getList 学生获得已选课程

- 请求方法: GET
- 请求url: /api/v1/student/course
- 请求功能描述: 学生获得所有已选课程。
- 入参传递方式: json
- 入参定义: 无入参
- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": success,  
4     "data": [  
5         {
```

```

6         "studentCourseId": 1,    // 选课表id
7         "courseName": "Math",
8         "teacherName": "一个老师",
9         "credit" : 2,
10        "score": 0,    // 总成绩
11    },
12    .....
13 ]
14 }
```

4.5 deleteCourse

- 请求方法: DELETE
- 请求url: /api/v1/student/course
- 请求功能描述: 学生退课
- 入参传递方式: json
- 入参定义:

```

1 {
2     "studentCourseId": 1, // 课程id
3 }
```

- 返回数据:无

```

1 {
2     "code": 0,
3     "msg": success,
4     "data": null
5 }
```

4.6 getExamList

- 请求方法: GET
- 请求url: /api/v1/student/exam

- 请求功能描述：学生考试信息查询
- 入参传递方式：json
- 入参定义：

```
1 {  
2     无入参  
3 }
```

- 返回数据：

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4     "data": [  
5         {  
6             "studentCourseId": 1,    // 选课id  
7             "courseName": "Math",  
8             "teacherName": "一个老师",  
9             "examDate": "yyyy-MM-dd HH:mm",    // 考试日期  
10            "examLocation": "一个地点",    // 考试地点  
11        },  
12    ]  
}
```

4.7 getStudentInfo

- 请求方法: GET
- 请求url: /api/v1/student/info
- 请求功能描述：学生个人信息查询
- 入参传递方式：json
- 入参定义：

```
1 {  
2     无入参  
3 }
```

- 返回数据:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": 20373903,    // 学号
7             "name": "xxx",
8             "departmentName": "cs", // 系名
9             "majorName": "?",    // 专业名
10            "className": "11",
11            "email": "xxxx@qq.com",
12            "birthday": "2002-01-01",
13            "sex": 0,    // 0 或 1
14            "password": 123456,
15        },
16    ]
}
```

4.8 updateStudentInfo

- 请求方法: PUT
- 请求url: /api/v1/student/info
- 请求功能描述: 学生个人信息更新
- 入参传递方式: json
- 入参定义:

```
1
2 {
3     "id": "20373903",    // 学号
4     "name": "xxx",
5     "departmentName": "cs", // 系名
6     "majorName": "?",    // 专业名
7     "className": "11",
8     "email": "xxxx@qq.com",
9     "birthday": "2002-01-01",
10    "sex": 0,    // 0 或 1
11    "password": 123456,
12 }
```

- 返回数据:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

4.9 getToEvaluateCourseInfo

- 请求方法: GET
- 请求url: /api/v1/student/evaluateCourse
- 请求功能描述: 获得【待评价】的课程列表（返回值不包含已评价的课程）
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": "123123", //学号
3 }
```

- 返回数据:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "courseId": "001", //课程号
7             "courseName": "Math", //课程名
8             "teacherName": "李老师", //任课教师
9             "credit": 2, //学分
10        },
11        .....
12    ]
13 }
```

4.10 submitEvaluation

- 请求方法: PUT
- 请求url: /api/v1/student/evaluateCourse
- 请求功能描述: 上传评价
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": "123123", //学号  
3     "courseId": "001", //被评价的课程号  
4     "star": 4, //评分的星级, 范围为1~5  
5     "text": "孩子已经吃了三箱了", //评价的文字  
6 }
```

- 返回数据:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

4.11 getTimeTable 【】

- 请求方法: GET
- 请求url: /api/v1/student/timetable
- 请求功能描述: 返回该学生的课程表。（data是一个[7][10]的二维列表）
- 入参传递方式: json
- 入参定义:

```
1 {  
2     无入参  
3 }
```

- 返回数据格式:

```
1 {
```

```

2     "code": 0,
3     "msg": "success",
4     "data": [
5         [
6             {
7                 "courseName": "",
8                 "teacherName": "",
9                 "location": "",
10            },
11            .....
12        ],
13        .....
14    ]
15 }

```

Admin:

1. 专业管理(AdminMajor) done

1.1 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/major/page/count
- 请求功能描述: 根据系名+专业名, 返回筛选过的所有专业页面数。
- 入参传递方式: json
- 入参定义:

```

1 {
2     "departmentName": "xxxxx", // 所属系名
3     "name": "xxxxxx" // 专业名
4     "pageSize": 6,
5 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5     {

```

```
6         "count": 12, // 总页面数
7     },
8 }
```

1.2 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/major/page/:index
- 请求功能描述: 根据系名+专业名+访问页index, 返回筛选过的所有专业。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "index": 1, // 第几页, 从1开始
3     "departmentName": "xxxxx",
4     "name": "xxxxxxx" // 专业名
5     "pageSize": 6,
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 专业id
7             "name": "", // 专业名
8             "departmentName": "", // 所属系名
9         },
10        .....
11    ]
12 }
```

1.3 getAMajor

- 请求方法: GET
-

- 请求url: `/api/v1/admin/major/:id`
- 请求功能描述: 获取一条专业信息 (用于编辑操作)
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 专业id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5     {
6         "id": "001", // 专业id
7         "name": "", // 专业名
8         "departmentName": "", // 所属系名
9     },
10 }
```

1.4 modifyAMajor

- 请求方法: PUT
- 请求url: `/api/v1/admin/major`
- 请求功能描述: 修改专业
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": "-1", // 专业id
3     "name": "", // 专业名
4     "departmentName": "", // 所属系名
5 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

1.5 createAMajor

- 请求方法: POST
- 请求url: /api/v1/admin/major
- 请求功能描述: 创建一个新专业 (id值固定为-1, 应由后端自动创建一此对象的id)
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": "-1", // 专业id  
3     "name": "", // 专业名  
4     "departmentName": "", // 所属系名  
5 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

1.6 deleteAMajor

- 请求方法: DELETE
- 请求url: /api/v1/admin/major/:id
- 请求功能描述: 删除一个专业
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 专业id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

1.7 getListName

- 请求方法: GET
- 请求url: /api/v1/admin/major/names
- 请求功能描述: 获取所有专业名字列表。
- 入参传递方式: json
- 入参定义:

```
1 {
2     无入参
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 专业id
7             "name": "", // 专业名
8         },
```

```
9      .....
10    ]
11 }
```

2. 系管理(AdminDepartment) done

2.1 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/department/page/count
- 请求功能描述: 根据系名, 返回筛选过的所有系的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "name": "xxxxxx", // 系名
3     "pageSize": 6,
4 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "count": 12, // 总页面数
7         },
8 }
```

2.2 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/department/page/:index
- 请求功能描述: 根据系名+访问页index, 返回筛选过的所有系。
- 入参传递方式: json

- 入参定义:

```
1 {
2     "index": 1,    // 第几页, 从1开始
3     "name": "xxxxxx" // 系名
4     "pageSize": 6,
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 系id
7             "name": "", // 系名
8             "majorCount": 2, // 专业数
9             "teacherCount": 50, // 教师数
10        },
11        .....
12    ]
13 }
```

2.3 getADepartment

- 请求方法: GET
- 请求url: /api/v1/admin/department/:id
- 请求功能描述: 获取一条系信息 (用于编辑操作)
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 系id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "id": "001", // 系id
7             "name": "", // 系名
8             "majorCount": 2, // 专业数
9             "teacherCount": 50, // 教师数
10        },
11 }
```

2.4 modifyADepartment

- 请求方法: PUT
- 请求url: /api/v1/admin/department
- 请求功能描述: 修改系
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": "-1", // 系id
3     "name": "", // 系名
4     "majorCount": 2, // 专业数
5     "teacherCount": 50, // 教师数
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

2.5 createADepartment

- 请求方法: POST
- 请求url: /api/v1/admin/department
- 请求功能描述: 创建一个新系 (id值固定为-1, 应由后端自动创建一此对象的id)
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": "-1", // 系id  
3     "name": "", // 系名  
4     "majorCount": 2, // 专业数  
5     "teacherCount": 50, // 教师数  
6 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

2.6 deleteADepartment

- 请求方法: DELETE
- 请求url: /api/v1/admin/department/:id
- 请求功能描述: 删除一个系
- 入参传递方式: json
- 入参定义:

```
1 {  
2     params: {  
3         "id": "001", // 系id  
4     }  
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

2.7 getListName

- 请求方法: GET
- 请求url: /api/v1/admin/department/names
- 请求功能描述: 获取所有系名字列表。
- 入参传递方式: json
- 入参定义:

```
1 {
2     无入参
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 系id
7             "name": "", // 系名
8         },
9         .....
10    ]
11 }
```

3. 学生管理(AdminStudent) done

3.1 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/student/page/count
- 请求功能描述: 根据学生名+专业名+班级名, 返回筛选过的所有学生的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "majorName": "xxxxxx", // 专业名  
3     "className": "xxxxxx", // 班级名  
4     "name": "小红", // 人名  
5     "pageSize": 6,  
6 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4     "data":  
5         {  
6             "count": 12, // 总页面数  
7         },  
8 }
```

3.2 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/student/page/:index
- 请求功能描述: 根据学生名+专业名+班级名+访问页index, 返回筛选过的所有学生。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "index": 1, // 第几页, 从1开始  
3     "majorName": "xxxxxx", // 专业名  
4     "className": "xxxxxx", // 班级名
```

```
5     "name": "小红", // 人名
6     "pageSize": 6,
7 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 学生id(学号)
7             "name": "", // 学生姓名
8             "className": "", // 班级名
9             "majorName": "", // 专业名
10        },
11        .....
12    ]
13 }
```

3.3 getAStudent

- 请求方法: GET
- 请求url: /api/v1/admin/student/:id
- 请求功能描述: 获取一条学生信息 (用于编辑操作)
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 学生id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
```

```

3     "msg": "success",
4     "data":
5         {
6             "id": "001", // 学生id(学号)
7             "name": "", // 学生姓名
8             "className": "", // 班级名
9             "majorName": "", // 专业名
10            "password": "",
11            "email": "",
12            "birthday": "yyyy-MM-dd",
13            "sex": 0, // 或者1
14
15        },
16    }

```

3.4 modifyAStudent

- 请求方法: PUT
- 请求url: /api/v1/admin/student
- 请求功能描述: 修改学生
- 入参传递方式: json
- 入参定义:

```

1 {
2     "id": "001", // 学生id(学号)
3     "name": "", // 学生姓名
4     "className": "", // 班级名
5     "majorName": "", // 专业名
6     "password": "",
7     "email": "",
8     "birthday": "yyyy-MM-dd",
9     "sex": 0, // 或者1
10 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4 }

```

3.5 createAStudent

- 请求方法: POST
- 请求url: /api/v1/admin/student
- 请求功能描述: 创建一个新学生 (id值固定为-1, 应由后端自动创建一此对象的id)
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": "001", // 学生id(学号)
3     "name": "", // 学生姓名
4     "className": "", // 班级名
5     "majorName": "", // 专业名
6     "password": "",
7     "email": "",
8     "birthday": "yyyy-MM-dd",
9     "sex": 0, // 或者1
10 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

3.6 deleteAStudent

- 请求方法: DELETE
- 请求url: /api/v1/admin/student/:id
- 请求功能描述: 删除一个系
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 系id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

3.7 getListName

- 请求方法: GET
- 请求url: /api/v1/admin/student/names
- 请求功能描述: 获取所有学生名字列表。
- 入参传递方式: json
- 入参定义:

```
1 {
2     无入参
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 学生id
7             "name": "", // 学生名
8         },
```

```
9      .....
10     ]
11 }
```

3.8 导入学生表

url : /api/v1/admin/upload/studentTable

4. 学生选课登记册(AdminStudentCourse) 【这里主要用来查成绩】

4.1 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/student/course/page/count
- 请求功能描述: 根据学生名+课程名+班级名, 返回筛选过的所有学生成绩的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "studentName": "小红", // 学生名
3     "className": "xxxxxx", // 班级名
4     "courseName": "", // 课程名
5     "pageSize": 6,
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5     {
6         "count": 12, // 总页面数
7     },
8 }
```

4.2 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/student/course/page/:index
- 请求功能描述: 根据学生名+课程名+班级名+访问页index, 返回筛选过的所有学生成绩。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "index": 1,    // 第几页, 从1开始  
3     "studentName": "小红", // 学生名  
4     "className": "xxxxxx", // 班级名  
5     "courseName": "", // 课程名  
6     "pageSize": 6,  
7 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4     "data": [  
5         {  
6             "id": 1, // 选课id  
7             "studentName": "", // 学生姓名  
8             "courseName": "", // 课程名  
9             "majorName": "", // 专业名  
10            "className": "", // 学生班级名  
11            "score": 30, // 总分  
12        },  
13        .....  
14    ]  
15 }
```

4.3 getAScore

- 请求方法: GET
- 请求url: /api/v1/admin/student/course/id

- 请求功能描述：获取一条学生成绩信息（用于编辑操作）
- 入参传递方式：json
- 入参定义：

```
1 {
2     params: {
3         "id": "001", // 选课id
4     }
5 }
```

- 返回数据格式：

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5     {
6         "id": "001", // 选课id
7         "studentName": "", // 学生姓名
8         "studentId": "20202727", // 学生学号
9         "courseName": "", // 课程名
10        "courseId": "023", // 课程id
11        "majorName": "", // 专业名
12        "className": "", // 学生班级名
13        "score": 30, // 总分
14    },
15 },
16 }
```

4.4 modifyAScore

- 请求方法: POST
- 请求url: /api/v1/admin/student/course
- 请求功能描述：修改记录
- 入参传递方式：json
- 入参定义：

```
1 {
```



```
2     "id": "-1", // 选课id
3     "studentId": "20202727", // 学生学号
4     "courseId": "023", // 课程id
5     "score": 30, // 总分
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

4.5 createAScore

- 请求方法: POST
- 请求url: /api/v1/admin/student/course
- 请求功能描述: 创建一个新记录 (id值固定为-1, 应由后端自动创建一此对象的id)
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": "-1", // 选课id
3     "studentId": "3", // 学生学号
4     "courseId": "16", // 课程id
5     "score": 30, // 总分
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

4.6 deleteAScore

- 请求方法: DELETE
- 请求url: /api/v1/admin/student/course/id
- 请求功能描述: 删除一条记录
- 入参传递方式: json
- 入参定义:

```
1 {  
2     params: {  
3         "id": "001", // 记录id  
4     }  
5 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

5. 管理员管理(AdminAdmin)

以下还在施工中，别做先

5.1 getAdminInfo

- 请求方法: GET
- 请求url: /api/v1/admin/admin/:id
- 请求功能描述: 根据管理员id，获得管理员的账户登录信息（用于编辑操作）;若没有提供id，则返回所有的管理员信息。
- 入参传递方式: json

- 入参定义:

```
1 //提供id
2 {
3     params: {
4         "id": "001", // 管理员id
5     }
6 }
7
8 //不提供id
9 {
10     无入参
11 }
```

- 返回数据格式:

```
1 //提供id
2 {
3     "code": 0,
4     "msg": "success",
5     "data":
6     {
7         "id": -1, // 管理员id
8         "username": "",
9         "privilege": "",
10        "password": ""
11    },
12 }
13
14 //不提供id
15 {
16     "code": 0,
17     "msg": "success",
18     "data": [
19         {
20             "id": -1, // 管理员id
21             "username": "",
22             "privilege": "",
23             "password": ""
24         },
25         .....
26     ]
27 }
28
```

5.2 createAdmin

- 请求方法: POST
- 请求url: /api/v1/admin/admin/
- 请求功能描述: 创建新的管理员
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": -1, // 管理员id  
3     "username": "",  
4     "privilege": "",  
5     "password": ""  
6 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success"  
4 }
```

5.3 deleteAdmin

- 请求方法: DELETE
- 请求url: /api/v1/admin/admin/:id
- 请求功能描述: 删除指定id的管理员
- 入参传递方式: json
- 入参定义:

```
1 {  
2     params: {  
3         "id": "001", // 管理员id  
4     }  
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

5.4 updateAdminInfo

- 请求方法: PUT
- 请求url: /api/v1/admin/admin/
- 请求功能描述: 更新管理员的信息
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": -1, // 管理员id
3     "username": "",
4     "privilege": "",
5     "password": ""
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

5.5 getAdminList

- 请求方法: GET
- 请求url: /api/v1/admin/admin/
- 请求功能描述: 获得管理员信息列表
- 入参传递方式: json
- 入参定义:

```
1 {
2     无入参
```

```
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success"
4     "data": [
5         {
6             "id": "002",
7             "username": "admin027",
8             "privilege": "255",
9             "password": "12345",
10        },
11        .....
12    ]
13 }
```

6. 班级管理(AdminClass) done

6.1 getClassInfo

- 请求方法: GET
- 请求url: /api/v1/admin/class/:id
- 请求功能描述: 根据班级id, 获得班级信息。
- 入参传递方式: json
- 入参定义:

```
1 //提供id
2 {
3     params: {
4         "id": "001", // 班级id
5     }
6 }
```

- 返回数据格式:

```
1 //提供id
2 {
3     "code": 0,
```

```
4     "msg": "success",
5     "data":
6     {
7         "id": -1, //班级id
8         "name": "", //班级名
9         "grade": 2019,
10        "departmentName": "xx系",
11        "majorName": "电竞",
12    },
13 }
14
```

6.2 createClass

- 请求方法: POST
- 请求url: /api/v1/admin/class/
- 请求功能描述: 创建新班级
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": -1, //班级id
3     "name": "", //班级名
4     "grade": 2019,
5     "departmentName": "xx系",
6     "majorName": "电竞",
7 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success"
4 }
```

6.3 deleteClass

- 请求方法: DELETE
- 请求url: /api/v1/admin/class/:id
- 请求功能描述: 删除指定id的班级

- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 班级id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

6.4 updateClassInfo

- 请求方法: PUT
- 请求url: /api/v1/admin/class/
- 请求功能描述: 更新班级的信息
- 入参传递方式: json
- 入参定义:

```
1 {
2     "id": -1, // 班级id
3     "name": "", // 班级名
4     "grade": 2019,
5     "departmentName": "xx系",
6     "majorName": "电竞",
7 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```


6.5 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/class/page/count
- 请求功能描述: 根据学生名+专业名+学院名, 返回筛选过的所有班级的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "majorName": "xxxxxx", // 专业名
3     "departmentName": "xxxxxx", // 学院名
4     "name": "小红", // 人名
5     "pageSize": 6,
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "count": 12, // 总页面数
7         },
8 }
```

6.6 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/class/page/:index
- 请求功能描述: 根据学生名+专业名+班级名+访问页index, 返回筛选得到的班级信息。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "index": 1, // 第几页, 从1开始
```

```

3     "majorName": "xxxxxx", // 专业名
4     "departmentName": "xxxxxx", // 学院名
5     "name": "小红", // 人名
6     "pageSize": 6,
7 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 班级id
7             "name": "", // 班级名
8             "grade": 2019, // 年级
9             "departmentName": "xxx", // 所属系
10            "majorName": "xxx", // 专业名
11        },
12        .....
13    ]
14 }

```

6.7 getMajorsName

- 请求方法: GET
- 请求url: /api/v1/admin/class/names
- 请求功能描述: 获得所有专业名以及专业id。
- 入参传递方式: json
- 入参定义:

```

1 {
2     无入参
3 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",

```

```

4      "data": [
5          {
6              "name": "计算机",
7              "id": "001"
8          },
9          .....
10     ]
11 }

```

6.8 getClassList

- 请求方法: GET
- 请求url: /api/v1/admin/class/name/list
- 请求功能描述: 获得所有班级信息列表
- 入参传递方式: json
- 入参定义:

```

1 {
2     无入参
3 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success"
4     "data": [
5         {
6             "id": "002", // 班级id
7             "name": "class027" // 班级名
8         },
9         .....
10    ]
11 }

```

7. 课程管理(AdminCourse)

7.1 getCourseInfo

- 请求方法: GET
- 请求url: /api/v1/admin/course/

- 请求功能描述：根据课程id，获得课程信息。
- 入参传递方式：json
- 入参定义：

```
1 //提供id
2 {
3     params: {
4         "id": "001", // 班级id
5     }
6 }
```

- 返回数据格式：

```
1 //提供id
2 {
3     "code": 0,
4     "msg": "success",
5     "data":
6         {
7             "id": -1,
8             "teacherId": null,
9             "name": "",
10            "grade": 2019,
11            "time": "",
12            "location": "",
13            "credit": 2,
14            "maxSize": 50,
15            "examDate": null,
16            "examLocation": null,
17        },
18 }
19
```

7.2 createCourse

- 请求方法: POST
- 请求url: /api/v1/admin/course/
- 请求功能描述：创建新课程
- 入参传递方式：json
- 入参定义：

```
1 {
2     "id": -1,
3     "teacherId": 1,
4     "name": "",
5     "grade": 2019,
6     "time": "",
7     "location": "",
8     "credit": 2,
9     "maxSize": 50,
10    "examDate": "",
11    "examLocation": "",
12 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success"
4 }
```

7.3 deleteCourse

- 请求方法: DELETE
- 请求url: /api/v1/admin/course/:id
- 请求功能描述: 删除指定id的课程
- 入参传递方式: json
- 入参定义:

```
1 {
2     params: {
3         "id": "001", // 课程id
4     }
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

7.4 updateCourseInfo

- 请求方法: PUT
- 请求url: /api/v1/admin/course/
- 请求功能描述: 更新班级的信息
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": -1, // 班级id  
3     "name": "终极一班", // 班级名  
4     "teacherName": "张三",  
5     "departmentName": "xx系",  
6     "grade": 2019,  
7     "credit": 4,  
8 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

7.5 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/course/page/count
- 请求功能描述: 根据课程名+专业名+学院名, 返回筛选过的所有课程。
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "index": 1, // 第几页, 从1开始  
3     "departmentName": "xxxxxx", // 学院名  
4     "name": "数据库", // 课程名  
5     "pageSize": 6,
```

```
6 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "count": 12, // 总页面数
7         },
8 }
```

7.6 getPage

- 请求方法: GET
- 请求url: `/api/v1/admin/course/page/:index`
- 请求功能描述: 根据课程名+专业名+学院名+访问页index, 返回筛选过的所有课程。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "index": 1, // 第几页, 从1开始
3     "majorName": "xxxxxx", // 专业名
4     "departmentName": "xxxxxx", // 学院名
5     "name": "数据库", // 课程名
6     "pageSize": 6,
7 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 课程id
7             "name": "", // 课程名
```

```

8         "teacherName": "", // 课程教师
9         "departmentName": "", // 所属系
10        "grade": 2019//年级
11        "credit":3//学分
12    },
13    .....
14 ]
15 }

```

7.7 getTeachersName

- 请求方法: GET
- 请求url: /api/v1/admin/course/names
- 请求功能描述: 获得所有教师名以及教师id。
- 入参传递方式: json
- 入参定义:

```

1 {
2     无入参
3 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "name": "陈老师",
7             "id": 1
8         },
9         .....
10    ]
11 }

```

7.8 getCourseList

- 请求方法: GET
- 请求url: /api/v1/admin/course/name/list

- 请求功能描述：获得所有课程信息列表
- 入参传递方式：json
- 入参定义:

```

1 {
2     无入参
3 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success"
4     "data": [
5         {
6             "id": "002", // 课程id
7             "name": "course027" // 课程名
8         },
9         .....
10    ]
11 }

```

8. 教师管理(AdminTeacher) done

8.1 getTeacherInfo

- 请求方法: GET
- 请求url: /api/v1/admin/teacher/:id
- 请求功能描述：根据课程id，获得教师信息。
- 入参传递方式：json
- 入参定义:

```

1 //提供id
2 {
3     params: {
4         "id": "001", // 班级id

```

```
5     }  
6 }
```

- 返回数据格式:

```
1 //提供id  
2 {  
3     "code": 0,  
4     "msg": "success",  
5     "data":  
6         {  
7             "id": -1, //教师id  
8             "number": "12312341234", //教师电话  
9             "name": "", //教师姓名  
10            "departmentName": "", //教师所属系  
11            "password": "" //密码  
12        },  
13 }  
14  
15
```

8.2 createTeacher

- 请求方法: POST
- 请求url: /api/v1/admin/teacher/
- 请求功能描述: 创建新教师
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": -1, //教师id  
3     "number": "12312341234", //教师电话  
4     "name": "", //教师姓名  
5     "departmentName": "", //教师所属系  
6     "password": "" //密码  
7 }
```

- 返回数据格式:

```
1 {
```

```
2     "code": 0,  
3     "msg": "success"  
4 }
```

8.3 deleteTeacher

- 请求方法: DELETE
- 请求url: /api/v1/admin/teacher/:id
- 请求功能描述: 删除指定id的教师
- 入参传递方式: json
- 入参定义:

```
1 {  
2     params: {  
3         "id": "001", // 教师id  
4     }  
5 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4 }
```

8.4 updateTeacherInfo

- 请求方法: PUT
- 请求url: /api/v1/admin/teacher/
- 请求功能描述: 修改某个教师的信息
- 入参传递方式: json
- 入参定义:

```
1 {  
2     "id": -1, //教师id  
3     "number": "12312341234", //教师电话  
4     "name": "", //教师姓名  
5     "departmentId": "", //教师所属系
```

```
6     "password": ""//密码
7 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4 }
```

8.5 getPageCount

- 请求方法: GET
- 请求url: /api/v1/admin/teacher/page/count
- 请求功能描述: 根据学院名+教师名, 返回筛选过的所有教师的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "departmentName": "xxxxxx", // 学院名
3     "name": "小红", // 人名
4     "pageSize": 6,
5 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "count": 12, // 总页面数
7         },
8 }
```

8.6 getPage

- 请求方法: GET
- 请求url: /api/v1/admin/teacher/page/:index
- 请求功能描述: 根据学院名+教师名+访问页index, 返回筛选过的所有教师。
- 入参传递方式: json
- 入参定义:

```

1 {
2     "index": 1,    // 第几页, 从1开始
3     "departmentName": "xxxxxx", // 学院名
4     "name": "小红", // 人名
5     "pageSize": 6,
6 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "001", // 教师id
7             "number": "123123" // 电话号码
8             "name": "", // 教师姓名
9             "departmentName": "", // 专业名
10        },
11        .....
12    ]
13 }

```

8.7 getDepartmentName

- 请求方法: GET
- 请求url: /api/v1/admin/teacher/names
- 请求功能描述: 获得所有系名与系id。
- 入参传递方式: json
- 入参定义:

```

1 {

```

```
2    无入参
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "name": "xx系", //系名
7             "id": "001" //系号
8         },
9         .....
10    ]
11 }
```

Teacher:

8. getTeacherCourse

- 请求方法: GET
- 请求url: /api/v1/teacher/course/list
- 请求功能描述: 获得该教师负责的课程的信息。
- 入参传递方式: json
- 入参定义:

```
1 {
2     无入参
3 }
```

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "id": "321", //课程id
```

```

7         "name": "数据库", //课程名
8         "grade": 2019, //年级
9         "credit": 3, //学分
10        "time": "3-3-2", // 星期几-第几节开始-上几节课
11        "location": "", //上课地点
12        "selectedCount": 2, //已选人数
13        "examDate": "yyyy-MM-dd HH:mm", //考试时间
14    },
15    .....
16 ]
17 }

```

9. getStudentGrade

- 请求方法: GET
- 请求url: /api/v1/teacher/grade/:id
- 请求功能描述: 根据选课id获得
- 入参传递方式: json
- 入参定义:

```

1 {
2     params: {
3         "id": "123" //选课id
4     }
5 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data": {
5         "studentCourseId": "321", //课程id (不是选课id!!), 这个地方参数名不
           是很合适, 但是不改比较方便了
6         "score": 100 //成绩
7     }
8 }

```

10. updateStudentGrade

- 请求方法: PUT
- 请求url: /api/v1/teacher/grade/:id
- 请求功能描述: 修改学生某门课的成绩
- 入参传递方式: json
- 入参定义:

```
1 {
2     "studentCourseId": "321", //课程id
3     "score": 100 //成绩
4 }
```

!!!!当前前端代码尚未传 "id": "312", //学生学号 这个参数!!!! 记得改在url上拼接

- 返回数据格式:

```
1 {
2     "code": 0,
3     "msg": "success" ,
4 }
```

11. getPageCount

- 请求方法: GET
- 请求url: /api/v1/teacher/grade/page/count
- 请求功能描述: 根据课程名+人名, 返回筛选过的所有成绩信息的页面数。
- 入参传递方式: json
- 入参定义:

```
1 {
2     "courseName": "xxxxxx", //课程名
3     "studentName": "小红", // 人名
4     "pageSize": 6,
5 }
```

- 返回数据格式:

```
1 {
```



```

2     "code": 0,
3     "msg": "success",
4     "data":
5         {
6             "count": 12, // 总页面数
7         },
8 }

```

5. getPage

- 请求方法: GET
- 请求url: /api/v1/teacher/grade/page/:index
- 请求功能描述: 根据学院名+教师名+访问页index, 返回筛选过的所有成绩信息。
- 入参传递方式: json
- 入参定义:

```

1 {
2     "index": 1, // 第几页, 从1开始
3     "courseName": "xxxxxx", //课程名
4     "studentName": "小红", // 人名
5     "pageSize": 6,
6 }

```

- 返回数据格式:

```

1 {
2     "code": 0,
3     "msg": "success",
4     "data": [
5         {
6             "studentCourseId": "001", // 课程id
7             "courseName": "123123" //课程名
8             "studentId": 1234 //学号
9             "studentName": "张三", // 学生姓名
10            "score": 100, // 成绩
11        },
12        .....
13    ]
14 }

```

6. getTimeTable

- 请求方法: GET
- 请求url: /api/v1/teacher/timetable
- 请求功能描述: 返回该教师的课程表。(data是一个[7][10]的二维列表)
- 入参传递方式: json
- 入参定义:

```
1 {  
2     无入参  
3 }
```

- 返回数据格式:

```
1 {  
2     "code": 0,  
3     "msg": "success",  
4     "data": [  
5         [  
6             {  
7                 "courseName": "",  
8                 "teacherName": "",  
9                 "location": "",  
10            },  
11            .....  
12        ],  
13        .....  
14    ]  
15 }
```