

课程管理 API 接口

增加课程接口

用来创建一个新的培训课程

请求语法

```
POST /api/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

action	必填	填写 add_course，表明是为了创建课程
data	必填	存储创建课程的信息，包括名称、描述、显示次序。 为 json 格式。例如：

```
{
  "name": "初中化学",
  "desc": "初中化学课程",
  "display_idx": "4"
}
```

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 必填 正常情况下该值将被设为 application/json，表示返回

JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示添加成功，id 是新加课程对应的 ID 号

```
{
  "retcode": 0
  "id": 1212
}
```

说明

增加课程的名称如果已经存在，则会创建失败返回结果为

```
{
  "retcode": 2,
  "reason": "同名课程已经存在"
}
```

列出课程接口

用来列出系统里所有的培训课程信息

请求语法

```
GET /api/mgr/sq_mgr/?action=list_course&pagenum=1&pagesize=20 HTTP/1.1
```

url 请求参数

action 填写 list_course，表明是要列出所有课程信息

pagenum 表示当前要显示的是第几页，目前固定填写 1

pagesize 表示一页最多显示多少条课程信息，目前固定填写 20

请求体内容

该请求无需指定请求内容。

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 正常情况下该值将被设为 application/json，表示返回 JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示

```
{
  "retlist": [
    {
      "desc": "初中语文",
      "id": 418,
      "display_idx": 1,
      "name": "初中语文"
    },
    {
      "desc": "初中数学",
      "id": 419,
      "display_idx": 2,
      "name": "初中数学"
    },
    {
      "desc": "初中英语",
      "id": 420,
      "display_idx": 3,
      "name": "初中英语"
    }
  ],
  "total": 3,
  "retcode": 0
}
```

retcode 值为 0 表示查询成功。

total 值表示总共有多少门课程信息

`retlist` 的内容是一个数组，其中每个元素对应一门课程信息。

修改课程接口

用来修改一门培训课程的信息

请求语法

```
PUT /api/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

action 填写 `modify_course`，表明是为了修改课程信息

id 填写要修改的课程的 `id` 号数字

newdata 存储创建修改后课程的信息，包括名称、描述、显示次序,为 `json` 格式。例如：

```
{
  "name": "初中化学",
  "desc": "初中化学课程",
  "display_idx": "4"
}
```

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 正常情况下该值将被设为 application/json，表示返回 JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示修改成功

```
{
  "retcode": 0
}
```

删除课程接口

用来删除一门培训课程

请求语法

```
DELETE /api/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

action	是	填写 delete_course，表明是为了删除课程信息
id	是	填写要删除的课程的 id 号

响应语法

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

头部信息

Content-Type 正常情况下该值将被设为 application/json，表示返回 JSON 格式的文本信息。

响应内容

如果删除成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示删除成功

```
{
  "retcode": 0
}
```

增加课程接口 2

除了上面说的增加课程接口，还提供请求体完全是 json 格式的方式添加课程

请求语法

```
POST /apijson/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/json
```

url 请求参数

无 url 请求参数

请求体内容

```
{
  "action" : "add_course_json",
  "data" : {
    "name": "初中化学",
    "desc": "初中化学课程",
  }
}
```

```
"display_idx": "4"
}
```

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 必填 正常情况下该值将被设为 application/json，表示返回 JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示添加成功

```
{
  "retcode": 0
}
```

说明

增加课程名称如果已经存在，则会创建失败返回结果为

```
{
  "retcode": 2,
  "reason": "同名课程已经存在"
}
```

老师管理 API 接口

增加老师接口

用来创建一个新的培训老师

请求语法

```
POST /api/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

action	必填	填写 add_teacher, 表明是为了创建老师
data	必填	存储创建老师的信息, 包括登录名、密码、真实姓名、描述信息、教授课程、显示次序。 为 json 格式。例如: { "username": "lishiming", "password": "sq888", "realname": "李世民", "desc": "李世民老师", "courses": [{"id": 419, "name": "初中数学"}, {"id": 420, "name": "初中英语"}], "display_idx": 1 }

注意 上面教授课程字段 `courses` 里面填写的是这个老师 教授的每门课程的 `id` 号 和 课程名称

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 必填 正常情况下该值将被设为 application/json，表示返回 JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示添加成功，id 表示创建的老师的 id 号

```
{
  "retcode": 0,
  "id": 143
}
```

说明

增加老师登录名如果已经存在，则会创建失败返回结果为

```
{
  "reason": "登录名 xxxx 已经存在",
  "retcode": 1
}
```

列出老师接口

用来列出系统里所有的老师信息

请求语法

```
GET /api/mgr/sq_mgr/?action=list_teacher&pagenum=1&pagesize=20 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

action 填写 `list_teacher`，表明是要列出所有老师信息

pagenum 表示当前要显示的是第几页，目前固定填写 `1`

pagesize 表示一页最多显示多少条老师信息，目前固定填写 `20`

请求体内容

该请求无需指定请求内容。

响应语法

`HTTP/1.1 200 OK`

`Content-Type: application/json`

头部信息

Content-Type 正常情况下该值将被设为 `application/json`，表示返回 JSON 格式的文本信息。

响应内容

如果请求成功，返回 json 格式的消息体，如下所示

```
{
  "retlist": [
    {
      "username": "lishiming",
      "realname": "李世民",
      "display_idx": 1,
      "courses": [
        {
          "course_id": 419
        },
        {
          "course_id": 420
        }
      ],
      "id": 143,
      "desc": "李世民老师"
    },
    {
      "username": "zhuyuanzhang",
```

```
    "realname": "朱元璋",
    "display_idx": 1,
    "courses": [
      {
        "course_id": 419
      },
      {
        "course_id": 420
      }
    ],
    "id": 144,
    "desc": "朱元璋老师"
  }
],
"total": 2,
"retcode": 0
}
```

retcode 值为 0 表示查询成功。

total 值表示总共多少个老师

retlist 的内容是一个数组，其中每个元素对应一位老师信息。

修改老师接口

用来修改一位培训老师的信息

请求语法

PUT /api/mgr/sq_mgr/ **HTTP/1.1**

Content-Type: application/x-www-form-urlencoded

url 请求参数

无 url 请求参数

请求体内容

action 填写 `modify_teacher`，表明是为了修改老师信息

id 填写要修改的老师的 `id` 号

newdata 存储创建修改后老师的信息，包括名称、描述、显示次序,为 `json` 格式。例如：

```
{
  "username": "lishiming",
  "password": "sq888",
  "realname": "李世民",
  "desc": "李世民老师",
  "courses": [{"id": 419, "name": "初中数学"}, {"id": 420, "name": "初中英语"}],
  "display_idx": 1
}
```

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 正常情况下该值将被设为 `application/json`，表示返回 `JSON` 格式的文本信息。

响应内容

如果请求成功，返回 `json` 格式的消息体，如下所示，`retcode` 值为 0 表示修改成功

```
{
  "retcode": 0
}
```

删除老师接口

用来删除一位培训老师

请求语法

```
DELETE /api/mgr/sq_mgr/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

action	是	填写 delete_teacher, 表明是为了删除老师信息
id	是	填写要删除的老师的 id 号

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type 正常情况下该值将被设为 application/json, 表示返回 JSON 格式的文本信息。

响应内容

如果删除成功, 返回 json 格式的消息体, 如下所示, retcode 值为 0 表示删除成功

```
{
  "retcode": 0
}
```

登录认证 API 接口

用来进行用户登录认证

请求语法

```
POST /api/mgr/loginReq HTTP/1.1
Content-Type: application/x-www-form-urlencoded
```

url 请求参数

无 url 请求参数

请求体内容

username 必填	填写用户名
password 必填	填写用户密码

响应语法

```
HTTP/1.1 200 OK
Content-Type: application/json
```

头部信息

Content-Type	必填	该字段值为 application/json，表示返回 JSON 格式的文本信息。
Set-Cookie	必填	该字段保存本次登录的 sessionId，比如： sessionId=89emkau5vhyg8vcwfwvq2pvr7ul2t5sc

响应内容

如果请求成功，返回 json 格式的消息体，如下所示，retcode 值为 0 表示登录认证成功

```
{
  "retcode": 0
}
```

说明

如果输入的用户名或者密码错误，则返回结果为

```
{  
  "retcode": 1,  
  "reason": "用户或者密码错误"  
}
```

松勤songqir