1. 注意

由于学校服务器不支持PUT 和DELETE操作，所以PUT 和DELETE并入POST ,请求时以 request_method:POST/PUT/DELETE作为区分

2.错误信息汇总:

```
status 400，已存在(比如：用户名，角色名，节点名称，新闻属性和标签唯一，重复时返回此错误)

status 404 Not founded
status 401 Unauthorized
```

3.用户模块

GET /api/user/users

```
[{
id:int
user_name:str,
login_time:, （上一次登陆时间）
login_ip:str,( 上一次登陆ip)
status:bool( 该用户状态True正常，false被禁),
email:str,
nodes:[
{   id:int,
    node_name:str,
    status:bool,
    level:int
}]
roles:[{                 是roles不是role
id:int ,
role_name:str,
status:bool,
}
clubs:[
{
id: int,
club_name: str
    }]
]

}
]
```

POST /api/user/users

```
{
user_name:str,
password:str
email:str
role_name:list(str)         注意是list
clubs: list(str)
}
```

GET /api/user/users/id

```
{
int: int   ,                新添加的，其实也可以不要，因为与路由的id一致,格式统一一下加上吧
 user_name:str,
login_time:,（上一次登陆时间）
login_ip:str,( 上一次登陆ip)
status:bool( 该用户状态True正常，false被禁),
email:str,
nodes:[
{    id:int,
     node_name:str,
    status:bool,
    level:int
}]
roles:[{                     是roles不是role
id:int ,
role_name:str,
status:bool,
}
]
clubs:[
{
id: int,
club_name: str
    }]

}
```

PUT /api/user/users/id

```
{
user_name:str,
password:str
email:str
role_name:list(str)              注意是list
status:bool,
clubs: list(str)



} optional
```

DELETE /api/user/users/id

GET /api/user/current-user

```
{
    和 Get/api/user/users/id  返回的数据一样
}
```

GET /api/user/nodes

```
[
{
id: int
node_name:str,
status:bool,
level:int
}
]
```

GET /api/user/nodes/id

```
{
id: int
node_name:str,
status:bool,
level:int
}
```

PUT /api/user/nodes/id

```
{
status :bool
} optional
```

GET /api/user/roles

```
[{
id:int,
role_name:str,
status:bool,
nodes:[
    {
    id:int,
     node_name:str,
    status:bool,
    level:int
}
]
}]
```

POST /api/user/roles(新建一角色)

```
    {
    role_name:str
    node_name:list( 要添加的节点名)
    }
```

GET /api/user/roles/id

```
    {
    id:int ,
     role_name:str,
    status:bool,
    nodes:[
        {
        id: int,
         node_name:str,
        status:bool,
        level:int
    }
    ]
    }
```

PUT /api/user/roles/id

```
    {
    role_name:list(str),              注意是list
```

```
status:bool,
node_name:list
}
optional
```

DELETE /api/user/roles/id

4.搜索模块

GET /api/Search/news?category=&tags=&start=&end=&sort

```
{
    和 GET/api/news/news 返回的数据格式一样
}注:tags可以多个,start-end为搜索的时间范围,为时间戳  sort=latest/oldest  反向或正向排序
```

5. template

GET /api/templates/fp:path

> ("api/templates/admin/app.html" .eg)

6. 文章

GET /api/news/news?limit=&offset= GET /api/clubs//articles

```
->

{
    paging: {
        limit: int
        offset: int
        total: int
    }
    data:[{
        id: int,
        author: str,
        category: str,
        tags: [str],
        post_time: UTC timestamp seconds,
        title: str,
        outline: str,
        status: boolean,( 审核是否成功)
    }]
}
```

POST /api/news/news # 新闻文章 POST /api/clubs//articles #社团空间文章

```
{
    category: str,
    tags: [str],
    title: str,
    detail: str,
}
```

```
->
```

```
HTTP state code
```

```
200 success
```

GET /api/news/news/id:str GET /api/clubs//articles/ ->

```
{
    id: int,
    category: str,
    tags: [str],
    post_time: UTC timestamp seconds,
    title: str,
    outline: str,
    status: boolean,
    detail: str
}

or

404
```

PUT /api/news/news/id:str PUT /api/clubs//articles/

```
{
    category: str,
    tags: [str],
    title: str,
    outline: str,
    status: boolean,
    detail: str
}

# optional keys

->

HTTP state code

200 OK
```

DELETE /api/news/news/id:str DELETE /api/clubs//articles/

```
->

200 OK
500 Not Authorized

# slideshow
```

GET /api/news/slider-shows

```
->

{
    paging: {
        limit: int
        offset: int
        total: int
    }
    data: [{
        id: int,
        post_time: UTC timestamp sec,
        title: str,
        img_url: str,
        outline: str,
        status: bool,
        link: str,
    }]
}
```

POST /api/news/slider-shows

```
{
    title: str,
    img_url: str,
    outline: str,
    link: str,
}
```

PUT /api/news/slider-shows/id:str

```
{
    title: str,
    img_url: str,
    outline: str,
    link: str,
    #status: bool,
}

#optional keys
```

DELETE /api/news/slider-shows/id:str

```
->

200 OK
```

GET /api/article/tags

```
[{
    id:int,
    name:str
}]
```

POST /api/article/tags

```
{
    name:str
}
```

GET /api/article/tags/id

```
 {
    id:int,
    name:str
}
```

PUT /api/article/tags/id

```
{
    name：str
}
```

DELETE /api/article/tags/id

GET /api/article/categorys

```
[{
    id:int,
    name:str
}]
```

POST /api/article/categorys

```
{
    name:str
}
```

GET /api/article/categorys/id

```
 {
    id:int,
    name:str
}
```

PUT /api/article/categorys/id

```
{
    name
}
```

DELETE /api/article/categorys/id

社团

POST /api/clubs

```
{
    name: str
    brief_introduction: str
    category: str
    picture: (data url)
}
```

GET /api/clubs

```
[{
    id: int
    name: str
    brief_introduction: str
    category: str
    picture: str(picture link)
    }]
```

GET /api/clubs/id

```
{
    id: int
    name: str
    brief_introduction: str
    category: str
    picture: str(picture link)
}
```

DELETE /api/clubs/id

POST /api/clubs/id/introduction #用于社联页面每个社团的介绍

```
{
    content: (富文本)
}
```

PUT /api/clubs/id/introduction

```
{
    content
}
```

GET /api/clubs/id/introduction

```
{
    content
}
```

报名

POST /api/sign-up

```
{
name: str
gender: str
major: str
phone: str
first_choice: str
second_choice: str
is_adjust: bool
self_introduction: str
}
```

GET /api/sign-up

```
[{
id: int
name: str
gender: str
major: str
phone: str
first_choice: str
second_choice: str
is_adjust: bool
self_introduction: str
apply_time: time stamp
}]
```

GET /api/sign-up/id

```
{
id: int
name: str
gender: str
major: str
phone: str
first_choice: str
second_choice: str
is_adjust: bool
self_introduction: str
apply_time: time stamp
}
```

GET /api/sign-up/download(为一个包含所有报名者的docx文件)

DELETE /api/sign-up/id

资料站

POST /api/data-station

```
{
    file: (html 中的file类型)
}
```

GET /api/data-station?status=&is_important(status:0为未审核，１为通过，－１为未通过,2为全部, is_important:0为一般，　１为重要，２为所有) eg. status=2&is_important=1 用于首页重要文件展示

```
[{
id: int
file_name: str
uploader: str
upload_time: int(time stamp)
downlowd_times: int
}]
```

PUT /api/data-station/id

```
{
file_name: str
file
status(审核状态，0为未审核，１为通过，－１为未通过)
is_important: bool(0 为一般，　１为重要文件)
}
```

GET /api/data-station/id

```
{
id: int
file_name: str
uploader: str
upload_time: int(time stamp)
downlowd_times: int
}
```

GET /api/data-station/id/download

DELETE /api/data-station/id