接口文档

课程查询修改相关

path ：/querycourse

方法：post

参数：

# 同csv文件中顺序基本一致，时间处略有不同

token

'cid':'00100873' , #课程id，注意是字符串

‘name’:’…’, #支持模糊查询

'category':'任选', #四个下拉选项之一

'credit':3.0, #如果输入整数最好转成保留一位小数的形式，比如输入3就转成3.0

‘teacher’:’…’,

# 对于时间我想做成这种形式，周一至七做成复选框，选择这一天就传入true到字典中，否则传入false。每个复选框下有下拉选项，下拉具体时间段，目前想法是 1-2，3-4，5-6，7-8，10-11，不提供连续三节及以上的课的选项

'mon':true,

'moti':'1-2',

'tue':true,

'tuti':'1-2',

'wed':true,

'weti':'1-2',

'thu':true,

'thti':'1-2',

'fri':true,

'frti':'1-2',

'sat':true,

'sati':'1-2',

'sun':true,

'suti':'1-2',

'depart':'艺术学院' #下拉选项，跟选课网一样

描述：用于根据各种信息查询课程信息。带引号的都是string类型，这些参数均可填可不填，不填的话就不必将对应属性加入post表单，但不能全空（需要前端检查一下）

返回消息：见尾注

path：/admin/insertinfo

方法：post

参数：同上

描述：管理员增加课程

返回消息：成功返回success，失败返回fail

path：/admin/updateinfo

方法：post

参数：同上

描述：管理员更新课程信息

返回消息：成功返回success，失败返回fail

其他接口：

（方便起见，返回消息如果是

成功返回success，失败返回fail

则 留空）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| path | 方法 | 参数 | 描述 | 返回消息 |
| /admin/deleteinfo | post | token ,cid | 管理员根据课程号删除消息 |  |
| /comment/querybyuid | get | token | 根据uid查询评论 | comment（评论，string） |
| /comment/querybycid | get | token ,cid | 根据cid查询评论 | comid（评论id），  comment |
| /comment/submitcomm | post | token，  cid，  comment | 用户提交评论 |  |
| /comment/deletecomment | post | token ,comid | 用户删除评论 |  |
| /admin/deletecomment | post | token ,comid | 管理员删除评论 |  |
| /ddl/queryddl | post | token ,cid | 用户查询ddl，cid为空表示查询所有ddl，返回升序后结果  ddltime格式为  2020-09-05-14-58  ddlstate true表示已完成，false表示未完成 | ddlid（int）,  uid,  ddlcontent（string）,  ddltime（string）,  ddlstate（bool） |
| /ddl/insertddl | post | token ,cid，  ddlcontent，  ddltime，  ddlstate | 用户插入ddl条目 |  |
| /ddl/updatestate | post | token ,ddlid，  ddlstate | 用户改变ddl完成状态 |  |
| /ddl/updateddlinfo | post | token ,ddlid，  ddlcontent，  ddltime | 用户更改ddl信息 |  |
| /ddl/deleteddl | post | token ,ddlid | 用户删除ddl |  |
| /usrcou/insertusrcou | post | token ,cid | 用户增加自己的课程 |  |
| /usrcou/queryusrcou | post | token , | 用户查询自己的课程 | cid |
| /user/enroll | post | username,password | 用户登录 | Token(成功时)字符串（出现的不同类型错误） |
| /admin/enroll | post | username,password | 管理员登录 | Token(成功时)字符串（出现的不同类型错误） |
| /userinfo/one | post | username,password | 查看用户信息 | Json格式包含下列信息：success（查看信息这一操作的状态，即成功查询还是出现了不同类型错误）,id,username,password,email, rk |
| /userinfo/all | post |  | 查看全部用户信息 | json格式，字典的键为数字（当前数据库中用户顺序），值为另一个字典（其键值为/userinfo/one  的返回消息） |
| /user/alter | post | username,password,new\_password | 用户修改密码 | 有关token的错误信息（错误时）  字符串change password success（成功时） |
| /admin/alter | post | username,new\_password | 管理员修改密码 | 有关token的错误信息（错误时）  字符串admin change password success（成功时） |
| /admin/rank | post | username | 管理员提升权限 | 有关token的错误信息（错误时）  字符串admin change rank success  （成功时） |
| /admin/delete | post | username | 用户删除 | 有关token的错误信息（错误时）  字符串delete success(成功时) |
| /user/register | post | username,password,email | 用户注册 | this name have existed或register successs |
| /usrcou/deleteusrcou | post | id,cid | 用户删除自己的课程 | Token有关错误或delete success（成功时）、delete ddl fail  delete usrcoutable fail  、delete ddl and usrcoutable both fail |
| /recommend | Post | Token,type1,type2(课程类型，eg专业必修)，recom\_num（推荐的课程数） | 推荐课程 | Json例子如下  字符串是课号cid |
| /usrcou/oldinsertusrcou | post | token ,cid | 用户增加自己已经上过的课程 |  |
| /usrcou/oldqueryusrcou | post | token , | 用户查询自己已经上过的课程 | cid |
| /usrcou/olddeleteusrcou | post | id,cid | 用户删除自己已经上过的课程 |  |

有需要可以继续加

注：

使用http协议通信

向主机 47.92.240.179 5001端口发送 http 请求，

将参数封装 url 中使用 get 或封装在 post 表单中使用post方法得到；

示例：在服务器程序运行的情况下，主机使用 curl 工具输入命令：

curl -i -d 'name=计算机' http://47.92.240.179:5001/querycourse

就能得到服务器返回的 查询课程 的结果

其中/querycourse就是文档中的path，'name=计算机'就是文档中的post参数

java中有对应的基于socket的各种库用来封装http请求，我查到一个，可以参考：<https://blog.csdn.net/qq9808/article/details/78320816>

返回格式

{"56": {"category": "\u4e13\u4e1a\u5fc5\u4fee", "note": "", "examtime": "", "sat": "", "name": "\u8ba1\u7b97\u673a\u7cfb\u7edf\u5bfc\u8bba\u8ba8\u8bba\u73edSeminaronIntroductiontoComputerSystems", "cid": "04832363", "wed": "", "thu": "10--11", "totalweek": "1--16", "tue": "", "classnum": 6, "credit": 0.0, "mon": "", "chours": 32.0, "sun": "", "depart": "\u4fe1\u606f\u79d1\u5b66\u6280\u672f\u5b66\u9662", "fri": "", "teacher": "\u9648\u5411\u7fa4", "stucount": 25},……}

该结果是 json 格式，第一个 56 表示该条目是查询的第56条记录，56属性对应的字典是一条课程信息（由于json使用unicode码编码中文，所以这里显示unicode码，前端可能需要考虑解码问题），每一条课程信息均使用字典格式，即“key”：“value”的形式；

其他查询信息返回结果和课程查询有所不同，比如：

{"1": [2010, "hahhahahh", "???", 1], "0": [2010, "hahhahahh", "???", 0]}

除第一层是表示编号的字典之外，第二层的列表中按序存放查询返回的各属性

关于java中json格式，该文章可供参考：

<https://blog.csdn.net/xingfei_work/article/details/76572550>