

附录：接口规约

在该部分中，我们将呈现开发中前后端交互所涉及的全部接口，并呈现接口的基本信息（包括路径、方法与接口描述）、规约请求参数（Headers、Body和/或Query）以及返回数据示例。

上传头像

基本信息

Path: /user/uploadIcon

Method: PUT

接口描述: 用于实现用户头像的上传。需要注意限制大小（80k），我会验证图片格式，png jpg jpeg应该可以，但是可能会被过滤掉。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	multipart/form-data	是		

Body

参数名称	参数类型	是否必须	示例	备注
icon	file	是		大小需限制在80kb以内

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

修改密码

基本信息

Path: /user/updatePassword

Method: PUT

接口描述: 用于提供用户修改密码的能力。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

```
{
  "oldPassword": "123456",
  "newPassword": "1234"
}
```

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

修改用户信息

基本信息

Path: /user/update

Method: PUT

接口描述: 用于提供用户修改用户信息的能力。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

```
{
  "username": "qibatu",
  "email": "qibatu@outlook.com",
  "gender": "male",
  "universityID": 1,
  "majorID": 2,
  "birthday": "2018-09-09",
  "studentID": "1655555",
  "auth": 1,
  "admissionYear": "2016"
}
```

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

根据大学获得专业信息

基本信息

Path: /register/{universityID}

Method: GET

接口描述: 用于根据用户选择的大学，渲染该大学所属专业信息的列表。

请求参数

路径参数

参数名称	示例	备注
universityID	1	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": [
    {
      "id": 238,
      "universityID": 0,
      "majorID": null,
    }
  ]
}
```

```
    "majorName": "船舶与海洋工程"
  },
  {
    "id": 239,
    "universityID": 0,
    "majorID": null,
    "majorName": "交通运输"
  },
  {
    "id": 240,
    "universityID": 0,
    "majorID": null,
    "majorName": "土木工程"
  },
  {
    "id": 241,
    "universityID": 0,
    "majorID": null,
    "majorName": "工程力学"
  },
  {
    "id": 242,
    "universityID": 0,
    "majorID": null,
    "majorName": "海洋工程类"
  },
  {
    "id": 243,
    "universityID": 0,
    "majorID": null,
    "majorName": "机械类"
  },
  {
    "id": 244,
    "universityID": 0,
    "majorID": null,
    "majorName": "土木类"
  },
  {
    "id": 245,
    "universityID": 0,
    "majorID": null,
    "majorName": "机械工程"
  },
  {
    "id": 246,
    "universityID": 0,
    "majorID": null,
    "majorName": "能源与动力工程"
  },
  {
    "id": 247,
    "universityID": 0,
```

```
    "majorID": null,
    "majorName": "工业工程"
  },
  {
    "id": 248,
    "universityID": 0,
    "majorID": null,
    "majorName": "新能源科学与工程"
  },
  {
    "id": 249,
    "universityID": 0,
    "majorID": null,
    "majorName": "核工程与核技术"
  },
  {
    "id": 250,
    "universityID": 0,
    "majorID": null,
    "majorName": "机械类(国际化人才培养试点班)"
  },
  {
    "id": 251,
    "universityID": 0,
    "majorID": null,
    "majorName": "电子信息类"
  },
  {
    "id": 252,
    "universityID": 0,
    "majorID": null,
    "majorName": "自动化"
  },
  {
    "id": 253,
    "universityID": 0,
    "majorID": null,
    "majorName": "信息工程"
  },
  {
    "id": 254,
    "universityID": 0,
    "majorID": null,
    "majorName": "电子科学与技术"
  },
  {
    "id": 255,
    "universityID": 0,
    "majorID": null,
    "majorName": "计算机科学与技术"
  },
  {
    "id": 256,
```

```

    "universityID": 0,
    "majorID": null,
    "majorName": "测控技术与仪器"
  },
  {
    "id": 257,
    "universityID": 0,
    "majorID": null,
    "majorName": "电气工程及其自动化"
  },
  {
    "id": 258,
    "universityID": 0,
    "majorID": null,
    "majorName": "信息安全"
  },
  {
    "id": 259,
    "universityID": 0,
    "majorID": null,
    "majorName": "软件工程"
  },
  {
    "id": 260,
    "universityID": 0,
    "majorID": null,
    "majorName": "微电子科学与工程"
  },
  {
    "id": 261,
    "universityID": 0,
    "majorID": null,
    "majorName": "电子信息类(IEEE试点班)"
  },
  {
    "id": 262,
    "universityID": 0,
    "majorID": null,
    "majorName": "材料科学与工程"
  },
  {
    "id": 263,
    "universityID": 0,
    "majorID": null,
    "majorName": "环境科学与工程"
  },
  {
    "id": 264,
    "universityID": 0,
    "majorID": null,
    "majorName": "生物医学工程"
  }
]

```

```
    "id": 265,
    "universityID": 0,
    "majorID": null,
    "majorName": "航空航天工程"
  },
  {
    "id": 266,
    "universityID": 0,
    "majorID": null,
    "majorName": "化学类"
  },
  {
    "id": 267,
    "universityID": 0,
    "majorID": null,
    "majorName": "化学工程与工艺"
  },
  {
    "id": 268,
    "universityID": 0,
    "majorID": null,
    "majorName": "化学"
  },
  {
    "id": 269,
    "universityID": 0,
    "majorID": null,
    "majorName": "数学与应用数学"
  },
  {
    "id": 270,
    "universityID": 0,
    "majorID": null,
    "majorName": "物理学"
  },
  {
    "id": 271,
    "universityID": 0,
    "majorID": null,
    "majorName": "生物科学类"
  },
  {
    "id": 272,
    "universityID": 0,
    "majorID": null,
    "majorName": "生物技术"
  },
  {
    "id": 273,
    "universityID": 0,
    "majorID": null,
    "majorName": "生物工程"
  },
}
```

```
{
  "id": 274,
  "universityID": 0,
  "majorID": null,
  "majorName": "计算机科学与技术(人工智能方向)"
},
{
  "id": 275,
  "universityID": 0,
  "majorID": null,
  "majorName": "自动化(人工智能方向)"
},
{
  "id": 276,
  "universityID": 0,
  "majorID": null,
  "majorName": "信息工程(人工智能方向)"
},
{
  "id": 277,
  "universityID": 0,
  "majorID": null,
  "majorName": "自然保护与环境生态类"
},
{
  "id": 278,
  "universityID": 0,
  "majorID": null,
  "majorName": "资源环境科学"
},
{
  "id": 279,
  "universityID": 0,
  "majorID": null,
  "majorName": "动物科学"
},
{
  "id": 280,
  "universityID": 0,
  "majorID": null,
  "majorName": "植物科学与技术"
},
{
  "id": 281,
  "universityID": 0,
  "majorID": null,
  "majorName": "药学类"
},
{
  "id": 282,
  "universityID": 0,
  "majorID": null,
  "majorName": "药学"
```



```
},
{
  "id": 283,
  "universityID": 0,
  "majorID": null,
  "majorName": "临床药学(五年制)"
},
{
  "id": 284,
  "universityID": 0,
  "majorID": null,
  "majorName": "经济管理试验班"
},
{
  "id": 285,
  "universityID": 0,
  "majorID": null,
  "majorName": "金融学"
},
{
  "id": 286,
  "universityID": 0,
  "majorID": null,
  "majorName": "国际经济与贸易"
},
{
  "id": 287,
  "universityID": 0,
  "majorID": null,
  "majorName": "经济学"
},
{
  "id": 288,
  "universityID": 0,
  "majorID": null,
  "majorName": "工商管理"
},
{
  "id": 289,
  "universityID": 0,
  "majorID": null,
  "majorName": "会计学"
},
{
  "id": 290,
  "universityID": 0,
  "majorID": null,
  "majorName": "人力资源管理"
},
{
  "id": 291,
  "universityID": 0,
  "majorID": null,
```

```
    "majorName": "信息管理与信息系统"
  },
  {
    "id": 292,
    "universityID": 0,
    "majorID": null,
    "majorName": "法学"
  },
  {
    "id": 293,
    "universityID": 0,
    "majorID": null,
    "majorName": "英语"
  },
  {
    "id": 294,
    "universityID": 0,
    "majorID": null,
    "majorName": "日语"
  },
  {
    "id": 295,
    "universityID": 0,
    "majorID": null,
    "majorName": "德语"
  },
  {
    "id": 296,
    "universityID": 0,
    "majorID": null,
    "majorName": "汉语言文学(中外文化交流)"
  },
  {
    "id": 297,
    "universityID": 0,
    "majorID": null,
    "majorName": "行政管理"
  },
  {
    "id": 298,
    "universityID": 0,
    "majorID": null,
    "majorName": "传播学"
  },
  {
    "id": 299,
    "universityID": 0,
    "majorID": null,
    "majorName": "广播电视编导"
  },
  {
    "id": 300,
    "universityID": 0,
```

```
    "majorID": null,
    "majorName": "文化产业管理"
  },
  {
    "id": 301,
    "universityID": 0,
    "majorID": null,
    "majorName": "工业设计"
  },
  {
    "id": 302,
    "universityID": 0,
    "majorID": null,
    "majorName": "建筑学"
  },
  {
    "id": 303,
    "universityID": 0,
    "majorID": null,
    "majorName": "视觉传达设计(含环境设计)"
  },
  {
    "id": 304,
    "universityID": 0,
    "majorID": null,
    "majorName": "园林"
  },
  {
    "id": 305,
    "universityID": 0,
    "majorID": null,
    "majorName": "电子信息类(全英语授课)"
  },
  {
    "id": 306,
    "universityID": 0,
    "majorID": null,
    "majorName": "电子与计算机工程"
  },
  {
    "id": 307,
    "universityID": 0,
    "majorID": null,
    "majorName": "机械类(全英语授课)"
  },
  {
    "id": 308,
    "universityID": 0,
    "majorID": null,
    "majorName": "材料类(全英语授课)"
  },
  {
    "id": 309,
```

```
    "universityID": 0,
    "majorID": null,
    "majorName": "电子信息类(零法语基础,英语+法语授课)"
  },
  {
    "id": 310,
    "universityID": 0,
    "majorID": null,
    "majorName": "机械类(零法语基础,英语+法语授课)"
  },
  {
    "id": 311,
    "universityID": 0,
    "majorID": null,
    "majorName": "法语"
  },
  {
    "id": 312,
    "universityID": 0,
    "majorID": null,
    "majorName": "工科试验班类"
  },
  {
    "id": 313,
    "universityID": 0,
    "majorID": null,
    "majorName": "理科试验班类"
  },
  {
    "id": 314,
    "universityID": 0,
    "majorID": null,
    "majorName": "自然科学试验班(生命与环境交叉平台)"
  },
  {
    "id": 315,
    "universityID": 0,
    "majorID": null,
    "majorName": "统计学"
  },
  {
    "id": 316,
    "universityID": 0,
    "majorID": null,
    "majorName": "应用物理学"
  },
  {
    "id": 317,
    "universityID": 0,
    "majorID": null,
    "majorName": "天文学"
  },
  {
```

```
    "id": 318,
    "universityID": 0,
    "majorID": null,
    "majorName": "食品科学与工程"
  },
  {
    "id": 319,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学(本博连读)"
  },
  {
    "id": 320,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学(本博连读法语班)"
  },
  {
    "id": 321,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学"
  },
  {
    "id": 322,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学(儿科方向5+3)"
  },
  {
    "id": 323,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学(英语班)"
  },
  {
    "id": 324,
    "universityID": 0,
    "majorID": null,
    "majorName": "临床医学(本博连读口腔医学方向)"
  },
  {
    "id": 325,
    "universityID": 0,
    "majorID": null,
    "majorName": "口腔医学"
  },
  {
    "id": 326,
    "universityID": 0,
    "majorID": null,
    "majorName": "预防医学"
  },
  },
```

```
{
  "id": 327,
  "universityID": 0,
  "majorID": null,
  "majorName": "医学技术类"
},
{
  "id": 328,
  "universityID": 0,
  "majorID": null,
  "majorName": "医学检验技术"
},
{
  "id": 329,
  "universityID": 0,
  "majorID": null,
  "majorName": "食品卫生与营养学"
},
{
  "id": 330,
  "universityID": 0,
  "majorID": null,
  "majorName": "护理学"
}
]
```

注册

基本信息

Path: /register

Method: POST

接口描述: 用于实现用户注册的功能。重名时无法注册。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

```
{
  "username": "qibatu",
  "password": "123456",
  "email": "qibatu@outlook.com",
  "gender": "male",
  "universityID": 1,
  "majorID": 2,
  "birthday": "2018-09-09",
  "studentID": "1655555",
  "auth": 1,
  "admissionYear": "2016"
}
```

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

用户登出

基本信息

Path: /user/logout

Method: POST

接口描述: 用于实现用户登出系统的功能。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

登陆

基本信息

Path: /auth

Method: POST

接口描述：用于实现用户登录系统的功能。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

```
{
  "username": "qibatu",
  "password": "123456"
}
```

返回数据

```
{
  "token":
  "eyJhbGciOiJIUzUxMiJ9.eyJzdWUiOiJxawJhdHUiLCJleHAiOjE1Mzk1MjkyNjEsIm1hdCI6MTUzODkyNDQ2MjE0LjEi_6bJ8o0bzo_6oo9tOB80CCpI5NaQVc2ZUL65Kt1K1kmTyWBoG1MTiHpqNW1A3RLBn4jwmTcjGFQVI1CbAyyQ"
}
```

获得大学信息

基本信息

Path： /register/university

Method： GET

接口描述：用于返回所有可注册的大学列表。

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": [
    {
      "universityID": 1,
      "universityName": "同济大学"
    },
    {
      "universityID": 2,
```



```
    "universityName": "复旦大学"
  },
  {
    "universityID": 3,
    "universityName": "上海交通大学"
  }
]
```

获得头像

基本信息

Path: /user/getIcon

Method: GET

接口描述: 用于获取用户的头像。

获取我的收藏资源（根据收藏时间排序）

基本信息

Path: /resource/myFavourite/{pageID}

Method: GET

接口描述: 用于实现我的收藏页面渲染。

请求参数

路径参数

参数名称	示例	备注
pageID	1	分页从1开始。

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 3,
    "list": [
      {
        "resourceID": "5bd6dd6737bb13570da2d318",
        "id": 17,
```

```

        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
        "resourceName": "chubby",
        "description": " Google Chubby 是一个大名鼎鼎的分布式锁服务，GFS和Big Table等大型
系统都用他来解决分布式协作、元数据存储和Master选择等一系列与分布式锁服务相关的问题。Chubby的底层一致性
实现就是以Paxos算法为基础的。Chubby是一个面向松耦合分布式系统的锁服务，通常用于为一个由大量小型计算机构
成的松耦合分布式系统提供高可用的分布式锁服务。一个分布式锁服务的目的是允许他的客户端进程同步彼此的操作，
并对当前所处环境的基本状态信息达成一致。针对这个目的，Chubby提供了粗粒度的分布式锁服务，开发人员不需要使
用复杂的同步协议，而是直接调用Chubby的锁服务接口即可实现分布式系统中多个进程之间粗粒度的同步控制，从而保
证分布式数据的一致性。Chubby的客户端接口设计非常类似于UNIX文件系统结构，应用程序通过Chubby的客户端接
口，不仅能够对Chubby服务器上的整个文件进行读写操作，还能添加对文件节点的锁控制，并且能够订阅Chubby服务
端发出的一系列文件变动的事件通知。",
        "points": 3,
        "uploadTime": "2018-10-29 06:14:26"
    },
    {
        "resourceID": "5bdb1c223cd8755c778a8afb",
        "id": 19,
        "uploaderID": 13,
        "categoryID": 1,
        "resourceMajorID": 38,
        "resourceName": "naiad",
        "description": "Naiad is a distributed system for executing data parallel,
cyclic dataflow programs. It offers the high throughput of batch processors, the low
latency of stream ...",
        "points": 4,
        "uploadTime": "2018-11-01 11:30:51"
    },
    {
        "resourceID": "5bebfd149438d9de03d96c0e",
        "id": 21,
        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
        "resourceName": "spark",
        "description": "Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark
是UC Berkeley AMP lab (加州大学伯克利分校的AMP实验室)所开源的类Hadoop MapReduce的通用并行框架",
        "points": 4,
        "uploadTime": "2018-11-14 06:46:46"
    }
],
"pageNum": 0,
"pageSize": 3,
"size": 3,
"startRow": 1,
"endRow": 3,
"pages": 1,
"prePage": 0,
"nextPage": 1,
"isFirstPage": false,
"isLastPage": false,
"hasPreviousPage": false,

```

```
        "hasNextPage": true,
        "navigatePages": 8,
        "navigatepageNums": [
            1
        ],
        "navigateFirstPage": 1,
        "navigateLastPage": 1
    }
}
```

上传资源

基本信息

Path: /uploadResource

Method: POST

接口描述: 用于实现用户资源的上传功能。data为文件在mongodb的唯一序列号。会过滤文件，压缩文件，源文件等文本，office类，pdf可以，sh脚本，可执行文件等其他的应该都不行。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	multipart/form-data	是		
Authorization		是		

Body

参数名称	参数类型	是否必须	示例	备注
resource	file	是	backpropogation.pdf	<128M

返回数据

```
{
    "status": 200,
    "message": "OK",
    "data": "5bbf5bc8231f0216ddb21c3a"
}
```

下载资源

基本信息

Path： /downloadResource/{resourceID}

Method： GET

接口描述：用于完成资源的下载功能。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Authorization		是		
Content-Type	application/json	是		

路径参数

参数名称	示例	备注
resourceID	5bbf4ccb231f027854348615	

删除评论

基本信息

Path： /resource/comment/delete/{commentID}

Method： DELETE

接口描述：用于实现用户评论的删除。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

路径参数

参数名称	示例	备注
commentID	4	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

删除资源

基本信息

Path: /deleteResource/{resourceID}

Method: GET

接口描述: 用于实现对用户上传资源的删除。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Authorization		是		

路径参数

参数名称	示例	备注
resourceID	5bbf5bc8231f0216ddb21c3a	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

取消收藏

基本信息

Path: /resource/favourite/dislike/{resourceID}

Method: DELETE

接口描述：用于实现用户对已收藏资源取消的操作。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

路径参数

参数名称	示例	备注
resourceID	5bcb2992a7536f4c1f0513f0	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

取消点赞资源

基本信息

Path： /resource/suggest/undo/{resourceID}

Method： DELETE

接口描述：用于实现用户对已点赞的资源取消的操作。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

路径参数

参数名称	示例	备注
resourceID	5bebfd149438d9de03d96c0e	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

收藏资源

基本信息

Path: /resource/favourite/like/{resourceID}

Method: POST

接口描述: 用于实现用户对资源的收藏。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

路径参数

参数名称	示例	备注
resourceID	5bcb2992a7536f4c1f0513f0	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

资源的详细信息

基本信息

Path: /resource/detail/{resourceID}

Method: GET

接口描述：用于获取某资源的详细信息。

可能为空值，比如评论信息，表明没有过评论

favourite表明收藏

suggested表明点赞（踩也是suggested，但是值为0，不过应该不用知道自己踩了哪些吧）

点赞率=点赞数/（点赞数+踩数）

上传人平均分=他所有上传资源的平均分

上传人获赞率=他所有上传的资源的点赞数/（所有资源的点赞数+踩数）

请求参数

路径参数

参数名称	示例	备注
resourceID	5bf684209438d9f772c537fd	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "uploaderInfo": {
      "resourceID": "5bf684209438d9f772c537fd",
      "uploaderName": "chloe",
      "uploaderID": 13
    },
    "downloadTimes": 1,
    "favouriteNum": 1,
    "suggestInfo": {
      "suggestedNum": 1,
      "resourceSuggestedRate": 1
    },
    "commentInfo": {
      "commentNum": 8,
      "resourceAvgScore": 4.75
    },
    "uploaderAvgScore": 3,
    "uploaderSuggestedRate": 0.8
  }
}
```

点赞资源

基本信息

Path： /resource/suggest/make/{resourceID}

Method: POST

接口描述: 用于实现用户对资源的点赞。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/x-www-form-urlencoded	是		

路径参数

参数名称	示例	备注
resourceID	5bebfd149438d9de03d96c0e	资源id

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

用户对某个资源的历史记录

基本信息

Path: /resource/user/history/{resourceID}

Method: GET

接口描述: 用于获取某用户和资源交互的详细信息。

suggested 为: -1 表明没有赞/踩
0 表明踩过,
1表明赞过
favourite为: 0表明没有收藏过,
1表明收藏过

请求参数

路径参数

参数名称	示例	备注
resourceID	5bf684209438d9f772c537fd	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "suggested": -1,
    "favourite": 0
  }
}
```

获取我的下载资源（根据下载时间排序）

基本信息

Path: /resource/myDownload

Method: GET

接口描述: 用于渲染我的下载页面。

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 4,
    "list": [
      {
        "resourceID": "5bebfd149438d9de03d96c0e",
        "id": 21,
        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
        "resourceName": "spark",
        "description": "Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark 是UC Berkeley AMP lab（加州大学伯克利分校的AMP实验室）所开源的类Hadoop MapReduce的通用并行框架",
        "points": 4,
        "uploadTime": "2018-11-14 06:46:46"
      }
    ],
    "pageNum": 2,
    "pageSize": 3,
    "size": 1,
    "startRow": 4,
    "endRow": 4,
    "pages": 2,
    "prePage": 1,
    "nextPage": 0,
  }
}
```

```
    "isFirstPage": false,
    "isLastPage": true,
    "hasPreviousPage": true,
    "hasNextPage": false,
    "navigatePages": 8,
    "navigatepageNums": [
      1,
      2
    ],
    "navigateFirstPage": 1,
    "navigateLastPage": 2
  }
}
```

上传资源metadata

基本信息

Path: /uploadResourceMetaData

Method: PUT

接口描述: 用于实现上传资源的详细信息。

tags 最多五个，每个tag最多10个字符，空字符分开。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Authorization		是		
Content-Type	application/json	是		

Body

```
{
  "resourceID": "5bf794fb9438d91a388e7c53",
  "categoryID": 4,
  "resourceMajorID": 52,
  "resourceName": "Byzantine Generals Problem",
  "description": "拜占庭将军问题 (Byzantine Generals Problem) ，是由莱斯利·兰波特在其同名论文 [1] 中提出的分布式对等网络通信容错问题。在分布式计算中，不同的计算器通过通讯交换信息达成共识而按照同一套协作策略行动。但有时候，系统中的成员计算机可能出错而发送错误的信息，用于传递信息的通讯网络也可能导致信息损坏，使得网络中不同的成员关于全体协作的策略得出不同结论，从而破坏系统一致性。拜占庭将军问题被认为是容错性问题中最难的问题类型之一。",
  "points": 4,
  "tags": "consensus byzantine"
}
```

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```

获得我点赞的资源(根据点赞时间排序)

基本信息

Path： /resource/mySuggest/{pageID}

Method： GET

接口描述： 获取用户点赞过的全部资源。

请求参数

路径参数

参数名称	示例	备注
pageID	2	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 4,
    "list": [
```

```
{
  "resourceID": "5bebfd149438d9de03d96c0e",
  "id": 21,
  "uploaderID": 12,
  "categoryID": 4,
  "resourceMajorID": 52,
  "resourceName": "spark",
  "description": "Apache Spark 是专为大规模数据处理而设计的快速通用的计算引擎。Spark 是UC Berkeley AMP lab（加州大学伯克利分校的AMP实验室）所开源的类Hadoop MapReduce的通用并行框架",
  "points": 4,
  "uploadTime": "2018-11-14 06:46:46"
},
{
  "pageNum": 2,
  "pageSize": 3,
  "size": 1,
  "startRow": 4,
  "endRow": 4,
  "pages": 2,
  "prePage": 1,
  "nextPage": 0,
  "isFirstPage": false,
  "isLastPage": true,
  "hasPreviousPage": true,
  "hasNextPage": false,
  "navigatePages": 8,
  "navigatepageNums": [
    1,
    2
  ],
  "navigateFirstPage": 1,
  "navigateLastPage": 2
}
}
```

获得我的上传资源（根据上传时间排序）

基本信息

Path: /resource/myUpload/{pageID}

Method: GET

接口描述: 用于实现我的上传页面的渲染。

请求参数

路径参数

参数名称	示例	备注
pageID	3	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 10,
    "list": [
      {
        "resourceID": "5bcb2992a7536f4c1f0513f0",
        "id": 12,
        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
        "resourceName": "paxos",
        "description": "Paxos算法是莱斯利·兰伯特 (Leslie Lamport)1990年提出的一种基于消息传递的一致性算法。 这个算法被认为是类似算法中最有效的。Paxos 算法解决的问题是一个分布式系统如何就某个值（决议）达成一致。一个典型的场景是，在一个分布式数据库系统中，如果各节点的初始状态一致，每个节点执行相同的操作序列，那么他们最后能得到一个一致的状态。为保证每个节点执行相同的命令序列，需要在每一条指令上执行一个“一致性算法”以保证每个节点看到的指令一致。一个通用的一致性算法可以应用在许多场景中，是分布式计算中的重要问题。因此从20世纪80年代起对于一致性算法的研究就没有停止过。节点通信存在两种模型：共享内存（shared memory）和消息传递（Messages passing）。Paxos 算法就是一种基于消息传递模型的一致性算法。",
        "points": 5,
        "uploadTime": "2018-10-20 09:12:06"
      },
      {
        "resourceID": "5bcc1118fc39df4b05e9fa43",
        "id": 15,
        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
        "resourceName": "raft",
        "description": "Raft是一种共识算法，旨在替代Paxos。 它通过逻辑分离比Paxos更容易理解，但它也被正式证明是安全的，并提供了一些额外的功能。 Raft提供了一种在计算系统集群中分布状态机的通用方法，确保集群中的每个节点都同意一系列相同的状态转换。 它有许多开源参考实现，具有Go, C ++, Java和Scala中的完整规范实现。Raft通过当选的领导者达成共识。簇集群中的服务器是领导者或追随者，并且在选举的精确情况下可以是候选者（领导者不可用）。领导者负责将日志复制到关注者。它通过发送心跳消息定期通知追随者它的存在。每个追随者都有一个超时（通常在150到300毫秒之间），它期望领导者的心跳。接收心跳时重置超时。如果没有收到心跳，则关注者将其状态更改为候选人并开始领导选举。",
        "points": 2,
        "uploadTime": "2018-10-21 01:39:45"
      },
      {
        "resourceID": "5bd6dd6737bb13570da2d318",
        "id": 17,
        "uploaderID": 12,
        "categoryID": 4,
        "resourceMajorID": 52,
```

```
    "resourceName": "chubby",
    "description": " Google Chubby 是一个大名鼎鼎的分布式锁服务，GFS和Big Table等大型系统都用他来解决分布式协作、元数据存储和Master选择等一系列与分布式锁服务相关的问题。Chubby的底层一致性实现就是以Paxos算法为基础的。Chubby是一个面向松耦合分布式系统的锁服务，通常用于为一个由大量小型计算机构成的松耦合分布式系统提供高可用的分布式锁服务。一个分布式锁服务的目的是允许他的客户端进程同步彼此的操作，并对当前所处环境的基本状态信息达成一致。针对这个目的，Chubby提供了粗粒度的分布式锁服务，开发人员不需要使用复杂的同步协议，而是直接调用Chubby的锁服务接口即可实现分布式系统中多个进程之间粗粒度的同步控制，从而保证分布式数据的一致性。Chubby的客户端接口设计非常类似于UNIX文件系统结构，应用程序通过Chubby的客户端接口，不仅能够对Chubby服务器上的整个文件进行读写操作，还能添加对文件节点的锁控制，并且能够订阅Chubby服务端发出的一系列文件变动的事件通知。",
    "points": 3,
    "uploadTime": "2018-10-29 06:14:26"
  }
],
"pageNum": 3,
"pageSize": 3,
"size": 3,
"startRow": 7,
"endRow": 9,
"pages": 4,
"prePage": 2,
"nextPage": 4,
"isFirstPage": false,
"isLastPage": false,
"hasPreviousPage": true,
"hasNextPage": true,
"navigatePages": 8,
"navigatepageNums": [
  1,
  2,
  3,
  4
],
"navigateFirstPage": 1,
"navigateLastPage": 4
}
}
```

获得我的所有评论(按评论时间排序)

基本信息

Path: /resource/myComment/{pageID}

Method: GET

接口描述: 获取某用户全部的评论数据。

请求参数

路径参数

参数名称	示例	备注
pageID	1	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 2,
    "list": [
      {
        "id": 2,
        "userID": 13,
        "resourceID": "5bcc1118fc39df4b05e9fa43",
        "commentTitle": "test",
        "comment": "raft 写起来还是很简单的，严格参考第五节的图即可",
        "commentTime": "2018-11-14 04:06:23",
        "score": 3
      },
      {
        "id": 1,
        "userID": 13,
        "resourceID": "5bbf26ae231f020d3d004864",
        "commentTitle": "test",
        "comment": "我就试试看",
        "commentTime": "2018-11-14 04:03:41",
        "score": 3
      }
    ],
    "pageNum": 0,
    "pageSize": 3,
    "size": 2,
    "startRow": 1,
    "endRow": 2,
    "pages": 1,
    "prePage": 0,
    "nextPage": 1,
    "isFirstPage": false,
    "isLastPage": false,
    "hasPreviousPage": false,
    "hasNextPage": true,
    "navigatePages": 8,
    "navigatepageNums": [
      1
    ],
    "navigateFirstPage": 1,
    "navigateLastPage": 1
  }
}
```


获得所有资源分类

基本信息

Path: /resourceMajors

Method: GET

接口描述: 用于获取全部的资源分类。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Authorization		是		

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": [
    {
      "id": 86,
      "resourceMajorName": "临床医学"
    },
    {
      "id": 87,
      "resourceMajorName": "交通工程"
    },
    {
      "id": 75,
      "resourceMajorName": "会计学"
    },
    {
      "id": 73,
      "resourceMajorName": "传播学"
    },
    {
      "id": 70,
      "resourceMajorName": "俄语"
    },
    {
      "id": 81,
      "resourceMajorName": "信息安全"
    },
    {
      "id": 66,
```

```
    "resourceMajorName": "历史学"
  },
  {
    "id": 84,
    "resourceMajorName": "口腔医学"
  },
  {
    "id": 67,
    "resourceMajorName": "哲学"
  },
  {
    "id": 45,
    "resourceMajorName": "土木工程"
  },
  {
    "id": 61,
    "resourceMajorName": "地球物理学"
  },
  {
    "id": 46,
    "resourceMajorName": "地质工程"
  },
  {
    "id": 77,
    "resourceMajorName": "大气科学"
  },
  {
    "id": 63,
    "resourceMajorName": "工程力学"
  },
  {
    "id": 74,
    "resourceMajorName": "广告学"
  },
  {
    "id": 42,
    "resourceMajorName": "应用化学"
  },
  {
    "id": 44,
    "resourceMajorName": "建筑学"
  },
  {
    "id": 55,
    "resourceMajorName": "德语"
  },
  {
    "id": 57,
    "resourceMajorName": "护理学"
  },
  {
    "id": 39,
    "resourceMajorName": "数学"
```

```
},
{
  "id": 69,
  "resourceMajorName": "日语"
},
{
  "id": 72,
  "resourceMajorName": "朝鲜语"
},
{
  "id": 47,
  "resourceMajorName": "机械工程"
},
{
  "id": 78,
  "resourceMajorName": "材料化学"
},
{
  "id": 79,
  "resourceMajorName": "材料物理"
},
{
  "id": 76,
  "resourceMajorName": "核工程与核技术"
},
{
  "id": 54,
  "resourceMajorName": "汉语言"
},
{
  "id": 83,
  "resourceMajorName": "法学"
},
{
  "id": 68,
  "resourceMajorName": "法语"
},
{
  "id": 62,
  "resourceMajorName": "测绘工程"
},
{
  "id": 64,
  "resourceMajorName": "物理学"
},
{
  "id": 48,
  "resourceMajorName": "环境工程"
},
{
  "id": 59,
  "resourceMajorName": "生物科学"
},
},
```

```
{
  "id": 49,
  "resourceMajorName": "电子信息工程"
},
{
  "id": 51,
  "resourceMajorName": "电子科学与技术"
},
{
  "id": 50,
  "resourceMajorName": "电气信息工程"
},
{
  "id": 41,
  "resourceMajorName": "经济学"
},
{
  "id": 40,
  "resourceMajorName": "统计学"
},
{
  "id": 65,
  "resourceMajorName": "船舶工程"
},
{
  "id": 53,
  "resourceMajorName": "英语"
},
{
  "id": 82,
  "resourceMajorName": "药学"
},
{
  "id": 71,
  "resourceMajorName": "西班牙语"
},
{
  "id": 52,
  "resourceMajorName": "计算机科学与技术"
},
{
  "id": 60,
  "resourceMajorName": "车辆工程"
},
{
  "id": 38,
  "resourceMajorName": "软件工程"
},
{
  "id": 80,
  "resourceMajorName": "通信工程"
},
{
```

```
        "id": 43,
        "resourceMajorName": "金融学"
    },
    {
        "id": 85,
        "resourceMajorName": "预防医学"
    }
]
}
```

获得所有资源的类型

基本信息

Path: /resourceCategories

Method: GET

接口描述: 用于获取全部的资源类型。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Authorization		是		

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": [
    {
      "id": 6,
      "resourceCategoryName": "学生笔记"
    },
    {
      "id": 5,
      "resourceCategoryName": "教材"
    },
    {
      "id": 4,
      "resourceCategoryName": "科研文献"
    },
    {
      "id": 3,
      "resourceCategoryName": "考试试卷"
    },
  ],
}
```

```
{
  "id": 2,
  "resourceCategoryName": "课堂练习"
},
{
  "id": 1,
  "resourceCategoryName": "课堂讲义"
}
]
```

获得某个资源下的评论（根据时间排序）

基本信息

Path: /resource/comments/{resourceID}/{pageID}

Method: GET

接口描述: 用于渲染某个资源评论的列表，根据评论时间排序。

请求参数

路径参数

参数名称	示例	备注
resourceID	5bebfd149438d9de03d96c0e	
pageID	1	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 8,
    "list": [
      {
        "id": 15,
        "userID": 31,
        "resourceID": "5bf684209438d9f772c537fd",
        "commentTitle": "zab",
        "comment": "应该失败",
        "commentTime": "2018-11-23 08:08:38",
        "score": 5
      },
      {
        "id": 14,
```

```
        "userID": 31,
        "resourceID": "5bf684209438d9f772c537fd",
        "commentTitle": "zab",
        "comment": "应该失败",
        "commentTime": "2018-11-23 08:06:11",
        "score": 5
    },
    {
        "id": 13,
        "userID": 31,
        "resourceID": "5bf684209438d9f772c537fd",
        "commentTitle": "zab",
        "comment": "应该失败",
        "commentTime": "2018-11-23 08:05:54",
        "score": 5
    }
],
"pageNum": 0,
"pageSize": 3,
"size": 3,
"startRow": 1,
"endRow": 3,
"pages": 3,
"prePage": 0,
"nextPage": 1,
"isFirstPage": false,
"isLastPage": false,
"hasPreviousPage": false,
"hasNextPage": true,
"navigatePages": 8,
"navigatepageNums": [
    1,
    2,
    3
],
"navigateFirstPage": 1,
"navigateLastPage": 3
}
}
```

获得自己的上传资源的信息

基本信息

Path: /user/detail

Method: GET

接口描述: 用于渲染我的上传页面。

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "userID": 13,
    "username": "chloe",
    "myUploadNum": 4,
    "myDownloadNum": 5,
    "myFavouriteNum": 2,
    "mySuggestedNum": 3,
    "avgScore": 3,
    "suggestedRate": 0.8
  }
}
```

评论资源

基本信息

Path: /resource/comment/make

Method: POST

接口描述: 用于实现用户对某个资源的评论。

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

```
{
  "resourceID": "5bf7942c9438d91a388e7c4e",
  "comment": "我就测测看",
  "score": 5,
  "commentTitle": "zab"
}
```

返回数据


```
{
  "status": 200,
  "message": "OK",
  "data": {
    "id": 16,
    "userID": 35,
    "resourceID": "5bf7942c9438d91a388e7c4e",
    "commentTitle": "zab",
    "comment": "我就测测看",
    "commentTime": "2018-11-25 08:03:08",
    "score": 5
  }
}
```

资源推荐（根据分数排序）

基本信息

Path: /resource/recommend/{resourceMajorID}/{categoryID}/{pageID}

Method: GET

接口描述: 用于渲染资源详情页中的资源推荐模块。

同类资源按照分数排名，没评分不会搜到。

请求参数

路径参数

参数名称	示例	备注
resourceMajorID	52	
categoryID	4	
pageID	1	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 3,
    "list": [
      {
        "resourceID": "5bf684209438d9f772c537fd",
        "id": 22,
        "uploaderID": 13,

```

```
    "categoryID": 4,
    "resourceMajorID": 52,
    "resourceName": "zab",
    "description": "ZAB 协议是为分布式协调服务 zookeeper 专门设计的一种支持崩溃恢复的原子广播协议。在 zookeeper 中，主要依赖 ZAB 协议来实现分布式数据一致性，基于该协议，zookeeper 实现了一种主备模式的系统架构来保持集群中各个副本之间的数据一致性。",
    "points": 4,
    "uploadTime": "2018-11-22 06:25:42",
    "tags": "zab 一致性 数据 分布式 zookeeper"
  },
  {
    "resourceID": "5bcc1118fc39df4b05e9fa43",
    "id": 15,
    "uploaderID": 12,
    "categoryID": 4,
    "resourceMajorID": 52,
    "resourceName": "raft",
    "description": "Raft是一种共识算法，旨在替代Paxos。它通过逻辑分离比Paxos更容易理解，但它也被正式证明是安全的，并提供了一些额外的功能。Raft提供了一种在计算系统集群中分布状态机的通用方法，确保集群中的每个节点都同意一系列相同的状态转换。它有许多开源参考实现，具有Go, C ++, Java和Scala中的完整规范实现。Raft通过当选的领导者达成共识。簇集群中的服务器是领导者或追随者，并且在选举的精确情况下可以是候选者（领导者不可用）。领导者负责将日志复制到关注者。它通过发送心跳消息定期通知追随者它的存在。每个追随者都有一个超时（通常在150到300毫秒之间），它期望领导者的心跳。接收心跳时重置超时。如果没有收到心跳，则关注者将其状态更改为候选人并开始领导选举。",
    "points": 2,
    "uploadTime": "2018-10-21 01:39:45",
    "tags": "raft"
  },
  {
    "resourceID": "5bbf26ae231f020d3d004864",
    "id": 4,
    "uploaderID": 12,
    "categoryID": 4,
    "resourceMajorID": 52,
    "resourceName": "inceptionv4",
    "description": "微软亚洲研究院的何恺明在2015年提出了震惊业界的ResNet结构，这种结构和以往的Inception结构走了两条不同的道路：前者主要关注加大网络深度后的收敛问题，而Inception更关注特征维度上的利用。如果把这两种方法结合起来会有什么效果呢？Szegedy在2016年就试验了一把，把这两种 最顶尖的结构混合到一起提出了Inception-ResNet，它的收敛速度更快但在错误率上和同层次的Inception相同；Szegedy还对自己以前提出的Inception-v3进行了一番改良，提出了Inception-v4。。。",
    "points": 1,
    "uploadTime": "2018-09-08 19:19:19",
    "tags": "msra"
  }
],
"pageNum": 0,
"pageSize": 3,
"size": 3,
"startRow": 1,
"endRow": 3,
"pages": 1,
"prePage": 0,
"nextPage": 1,
```

```
        "isFirstPage": false,
        "isLastPage": false,
        "hasPreviousPage": false,
        "hasNextPage": true,
        "navigatePages": 8,
        "navigatepageNums": [
            1
        ],
        "navigateFirstPage": 1,
        "navigateLastPage": 1
    }
}
```

资源查找（根据分数排名）

基本信息

Path： /searchResource/score/{resourceMajorID}/{categoryID}/{pageID}?keyword=

Method： GET

接口描述： 用于实现资源查找的功能，需配合Elasticsearch。

请求参数

路径参数

参数名称	示例	备注
resourceMajorID	0	
categoryID	0	
pageID	2	

Query

参数名称	是否必须	示例	备注
keyword	是		

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
    "total": 6,
    "list": [
```

```
{
  "resourceID": "5bbf26ae231f020d3d004864",
  "id": 4,
  "uploaderID": 12,
  "categoryID": 4,
  "resourceMajorID": 52,
  "resourceName": "inceptionv4",
  "description": "微软亚洲研究院的何恺明在2015年提出了震惊业界的ResNet结构，这种结构和以往的Inception结构走了两条不同的道路：前者主要关注加大网络深度后的收敛问题，而Inception更关注特征维度上的利用。如果把这两种方法结合起来会有什么效果呢？Szegedy在2016年就试验了一把，把这两种最顶尖的结构混合到一起提出了Inception-ResNet，它的收敛速度更快但在错误率上和同层次的Inception相同；Szegedy还对自己以前提出的Inception-v3进行了一番改良，提出了Inception-v4。。。",
  "points": 1,
  "uploadTime": "2018-09-08 19:19:19",
  "tags": "msra"
},
{
  "resourceID": "5c0a12931a8709255c4022bf",
  "id": 59,
  "uploaderID": 72,
  "categoryID": 1,
  "resourceMajorID": 38,
  "resourceName": "计算机网络课件Lecture10",
  "description": "这是计算机网络Lecture10课件，由同济大学金伟祖老师呈现。",
  "points": 3,
  "uploadTime": "2018-12-07 02:26:33",
  "tags": "软件工程 计算机网络 课件 Lecture10"
},
{
  "resourceID": "5bdb1c223cd8755c778a8afb",
  "id": 19,
  "uploaderID": 13,
  "categoryID": 1,
  "resourceMajorID": 38,
  "resourceName": "naiad",
  "description": "Naiad is a distributed system for executing data parallel, cyclic dataflow programs. It offers the high throughput of batch processors, the low latency of stream ...",
  "points": 4,
  "uploadTime": "2018-11-01 11:30:51",
  "tags": "naiad"
}
],
"pageNum": 2,
"pageSize": 3,
"size": 3,
"startRow": 4,
"endRow": 6,
"pages": 2,
"prePage": 1,
"nextPage": 0,
"isFirstPage": false,
"isLastPage": true,
```

```
    "hasPreviousPage": true,
    "hasNextPage": false,
    "navigatePages": 8,
    "navigatepageNums": [
        1,
        2
    ],
    "navigateFirstPage": 1,
    "navigateLastPage": 2
}
}
```

资源查找（根据时间排序）

基本信息

Path： /searchResource/time/{resourceMajorID}/{categoryID}/{pageID}?keyword=

Method： GET

接口描述： 用于实现资源查找的功能，需配合Elasticsearch。

没有关键字就全文搜索

totalPages是总页数，size是一页大小 应该只有这个是从0开始的，最后一页=总页数-1

其他的分页0页和1页一样的，最后一页=总页数

请求参数

路径参数

参数名称	示例	备注
resourceMajorID	52	
categoryID	4	
pageID	1	

Query

参数名称	是否必须	示例	备注
keyword	是	zab	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": {
```

```

"content": [
  {
    "resourceID": "5bf684209438d9f772c537fd",
    "id": 22,
    "uploaderID": 13,
    "categoryID": 4,
    "resourceMajorID": 52,
    "resourceName": "zab",
    "description": "ZAB 协议是为分布式协调服务 zookeeper 专门设计的一种支持崩溃恢复的原子广播协议。在 zookeeper 中, 主要依赖 ZAB 协议来实现分布式数据一致性, 基于该协议, zookeeper 实现了一种主备模式的系统架构来保持集群中各个副本之间的数据一致性。",
    "points": 4,
    "uploadTime": "2018-11-22 06:25:42",
    "tags": "zab 一致性 数据 分布式 zookeeper"
  }
],
"pageable": {
  "sort": {
    "sorted": false,
    "unsorted": true
  },
  "offset": 0,
  "pageSize": 3,
  "pageNumber": 0,
  "paged": true,
  "unpaged": false
},
"facets": [],
"aggregations": null,
"scrollId": null,
"totalElements": 1,
"totalPages": 1,
"size": 3,
"number": 0,
"numberOfElements": 1,
"first": true,
"sort": {
  "sorted": false,
  "unsorted": true
},
"last": true
}
}

```

踩资源

基本信息

Path: /resource/suggest/dislike/{resourceID}

Method: GET

接口描述：用于实现踩资源的功能。

请求参数

路径参数

参数名称	示例	备注
resourceID	5bbe2d4d231f02a92961b5cd	

返回数据

```
{
  "status": 200,
  "message": "OK",
  "data": null
}
```