

提示：管理员不是用户！！

filter是虚的，拆开的话不留filter

所有的page都默认为1, page_size 默认为20

没有错误信息的返回的 error 为 None

1. 平台管理员 + 平台本身

设备租赁与平台的管理员，负责所有设备的申报审批、用户管理工作

1. 查询所有用户信息（可选是否只显示租借者）

```
Method: GET
URL: /api/user/query-all
QueryParam:
{
  'filter': 'None' / 'lessor' //设置是否只显示租借者信息
  'page':
  'page_size':
}
Response:
{
  'total': 213 //符合筛选条件的用户数量
  'users': [
    {
      "username": '',
      'student_id': ,
      'user_id': ,
      "email": '',
      "contact": '',
      "authority": '',
      'lab_info': ''
    }
  ]
  'error':
}
```

2. 设置某个用户的类型

```
Method: POST
url: /api/user/set-authority
Request:
{
  'user_id':
  'authority': 'user' / 'lessor'
}
Response
{
  'message': 'ok'
  'error': '' //有设备关联而不能删除该用户
}
```

3. 删除某个用户

```
Method: POST
url: /api/user/delete
Request
{
  'user_id':
}
Response
{
  'message' : 'ok'
  'error': '' //有设备关联而不能删除该用户
}
```

4. 查看所有设备的列表

```
Method: GET
url: /api/equip/query
QueryParam:
{
  'filter': { //可选传，由于是get，使用自行拆开
    'status': 'onsale' / 'rented' / 'unavailable'
    'lessor_name': '', //按照姓名筛选
    'lessor_id': , //按照出租者id进行筛选，非学号
    'name_search': //按照设备名进行筛选（搜索），
    'username':
    'user_id':
  }
  'page':
  'page_size':
}
Response:
{
  'total': 12 //符合筛选条件的设备数量
  'equip': [
    {
      'equip_id':
      'equip_name': ''
      'lessor_name': ''
      'address': ''
      'end_time': ''
      'contact': ''
      'status': 'onsale' / 'rented' / unavailable // 可租/已租出/未上架
      'username': ''
    }
  ]
  'error': ''
}
```

5. 修改设备信息:

```
Method: POST
url: api/equip/set
Request:
{
  'equip_id':
  'equip_name': ''
  'address': ''
}
```

```
    'end_time': ''
    'status': ''
  }
Response:
{
  'message': 'ok'
  'error': ''
}
```

6. 删除设备:

```
Method: POST
url: api/equip/delete
Request:
{
  'equip_id':
}
Response:
{
  'message': 'ok'
  'error': '' //可能会出现设备正出租而无法删除的情况
}
```

7. 管理员查看租借历史 (信息)

```
Method: GET
url: api/rent/query
QueryParam:
{
  'filter': {
    'equip_id':
    'lessor_id':
    'user_id':
    'equip_name': ''
    'lessor_name': ''
    'username': ''
  }
  'page_size':
  'page':
}
Response:
{
  'total':
  'rent_info': [
    {
      'rent_id':
      'equip_id':
      'equip_name': ''
      'lessor_name': ''
      'username': ''
      'rent_time': ''
      'return_time': '' //未归还为空串
      'end_time': ''
      'status': 'returned' / 'unreturned'
    }
  ]
}
```

```
]
  'error': ''
}
```

8. 查看租借申请

```
Method: GET
url: api/rent/request/query
QueryParam:
{
  'filter': { //可选传, 自行拆开+1
    'lessor_name': ''
    'lessor_id':
    'renter_name': ''
    'user_id':
    'equip_name': ''
    'equip_id':
  }
  'page_size':
  'page':
}
Response:
{
  'total':
  'rent_requests': [
    'rent_req_id':
    'equip_id':
    'equip_name': ''
    'lessor_name': ''
    'username': ''
    'start_time': ''
    'return_time': ''
    'detail': ''
    'status': 'apply' / 'reject' / 'pending'
  ]
  'error': ''
}
```

9. 查看上架申请:

```
Method: GET
url: api/equip/request/query
Request:
{
  'page':
  'page_size':
  'filter': { //可选传 自行拆开+2
    'lessor_name': ''
    'equip_name': ''
    'equip_id':
    'end_time': ''
  }
}
Response:
{
```

```
'total':
  'equip_req': [
    'sale_req_id':
    'equip_id':
    'equip_name': ''
    'end_time': ''
    'lab_info': ''
    'lessor_name': ''
    'status': 'apply' / 'reject' / 'pending'
  ]
  'error': ''
}
```

10. 管理员审核上架申请:

```
Method: POST
url: api/equip/request/decide
Request:
{
  'sale_req_id':
  'decision': 'apply' / 'reject'
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

11. 管理员审核租借申请:

```
Method: POST
url: api/rent/request/decide
Request:
{
  'rent_req_id':
  'decision': 'apply' / 'reject'
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

12. 删除租借申请:

```
Method: POST
url: api/rent/request/delete
Request:
{
  'rent_req_id':
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

13. 注册:

```
Method: POST
url: api/logon
Request:
{
  'username': ''
  'password': '' //前端请先加密
  'student_id': ''
  'email': ''
  'contact': ''
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

14. 登录

```
Method: POST
url: api/login
Request:
{
  'username': ''
  'password': ''
}
Response:
{
  'message': 'ok'
  'error': ''
}
登录cookie在响应头中, 名为'session_id'
```

15. 登出

```
Method: POST
url: api/logout
Request:
{

}
Response:
{
  'message': 'ok'
  'error': ''
}
```

16. 激活

```
Method: GET
url: api/active
QueryParam:
{
  'rand_str': ''
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

17. 获取当前登录的用户信息

```
Method: GET
url: api/user/info
Response:
{
  'user_id':
  'student_id':
  'username': ''
  'authority': ''
  'error': ''
}
```

18. 管理员审核普通用户变成设备提供者的审批

```
Method: POST
url: api/user/auth/decide
Request:
{
  'auth_req_id': ''
  'decision': 'apply' / 'reject'
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

19. 查看普通用户变成设备提供者的申请

```
Method: GET
url: api/user/auth/query
QueryParam:
{
  'filter': { //可选传 自行拆开
    'user_id': // 从1开始的那个用户id。不传此项表示查询所有的
  }
  'status': // 为'all' 或者 'pending', pending表示只查未审批的
  'page':
  'page_size':
}
Response:
{
  'total':
  'auth_req': [
    {
      'auth_req_id':
      'user_id':
      'username': ''
      'lab_info': ''
      'detail': ''
      'email':
      'contact':
      'status': ''
    }
  ]
}
```

2. 普通用户

1. 获取已上架设备列表、搜索：

1.4 使用 filter，自行拆开+3

2. 申请租借：

```
Method: POST
url: api/rent/request/add
request:
```



```
{
  'equip_id':
  'detail':
  'start_time':
  'return_time':
}
Response
{
  'message': 'ok'
  'error': ''
}
```

3. 查看自己的租借申请信息：

1.8中使用，后端会自行判断是不是admin，建议在filter中的'renter_name'加上相关信息

4. 查看自己的租借历史信息：

1.7中使用，后端会判断是否是管理员

5. 申请成为实验室助理：

```
Method: POST
url: api/user/auth/add
Request:
{
  'lab_info': ''
  'detail': ''
  'error': ''
}
```

3. 实验室工作助理（设备提供者）

1. 查看己方设备信息：

1.4 后端会判，但是前端建议写上filter

2. 修改己方设备信息：

1.5 后端会判

3. 删除己方设备：

1.6 后端会判

4. 增加设备：

```
Method: POST
url: api/equip/add
Request:
{
  'equip_name':
  'address':
}
Response:
{
  message: 'ok'
  'error': ''
}
```

5. 提交上架申请：

```
Method: POST
url: api/equip/request/add
Request:
{
  'equip_id':
  'end_time': ''
}
Response:
{
  'message': 'ok'
  'error': ''
}
```

6. 下架己方设备:

3.2

7. 审核租借申请

1.11

8. 查看已借出设备信息及其归还情况（租借历史）：

1.7 后端自己判 建议在 filter里面加上相关信息

9. 确认归还:

```
Method: POST
url: api/rent/confirm
Request:
{
  'rent_info_id':
}
Response:
{
  'message': 'ok'
  'error': ''
}
```