

dbook api 文档 v2

版本	备注	时间
1.0	1, 修改contract为token; 2, 出版api中, 参数token改为currency, 添加token信息	2022-11-26
1.1	修改外键book为issue	2022-11-27
1.2	修改购买api	2022-11-28
1.3	修改用户信息api { lowest_price -> min_price, highest_price -> max_price }	2022-12-01
1.4	取消draft和book的current api	2022-12-02
1.5	添加注释; 纠正个别api字段错误	2022-12-03
1.6	补充issue resale api	2022-12-05
1.7	添加常用流程	2022-12-11
1.8	添加NFT销毁流程	2022-12-16
1.9	添加时间说明	2022-12-24
2.0	补充用户信息statistic, 补充issue信息price_range, n_owned	2023-01-02

环境准备

0x1, 安装docker

1 参考<https://docs.docker.com/engine/install/>

0x2, 安装docker-compose(windows跳过此步骤)

1 参考<https://docs.docker.com/compose/install/>

0x3, 启动server

```
1 cd backend //进入项目根目录
2 docker-compose up --build
```

0x4, 关闭server

```
1 cd backend //进入项目根目录
2 docker-compose down
```

0x5, 访问api doc

```
1 http://localhost:8000/api-doc //登录用户: admin/admin
```

0x6, api登录认证

```
1 headers: {
2   "authorization": "Bearer {token}"
3 }
```

测试环境

```
1 api uri: http://dbookmarket.com:56789/api/v1/
2 api doc: http://dbookmarket.com:56789/api-doc/
3
4 后台管理
5 url: http://dbookmarket.com:56789/d-book-admin/
6 用户: admin/admin
```

接口说明

HTTP状态码

200: 返回成功

201: 创建成功

400: 表单验证报错

错误格式:

```
1 1, 字段验证报错
2 {
3   name: [error message],
4   desc: ...,
5   ...
6 }
7 2, 非字段验证报错
8 {
9   non_field_errors: []
10 }
11 3, 其他错误
12 {
13   detail: ''
14 }
```

401: 无法登陆

403: 无访问权限

405: 请求方法不允许

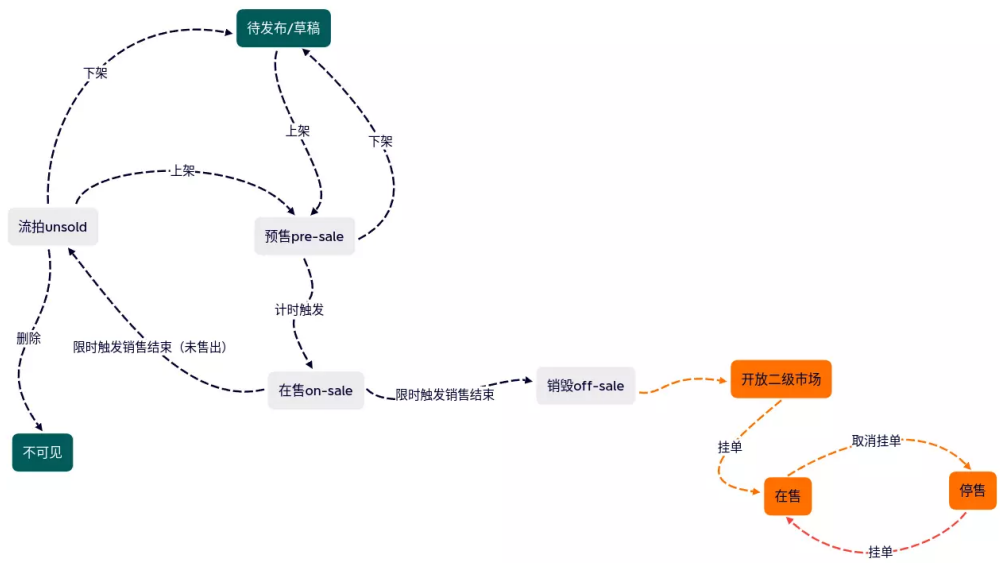
500: 系统错误

时间转换

系统返回的是UTC时间，前端需要转换为当地时间

```
1 //示例: moment.js
2 moment.utc("2022-11-29 03:15:00").local().format('YYYY-MM-DD HH:mm:ss')
```

书籍销售状态转移图



用户权限

角色	权限	级别（高等级包含低等级权限）
作者	添加draft, book, issue	1
其他	添加trade, transaction, wishlist, fans	2

常用流程

作者发布书籍流程

Step 1, 调用NFT合约setApprovalForAll()进行授权给platform合约，若成功，转step 2；
Step 2, 调用issue接口发布书籍。

用户购买首发书籍流程

Step 1, 调用platform合约payFirstTrade()转账给平台钱包，若成功，转step 2；
Step 2, 调用transaction接口购买书籍。

用户购买二手书籍

Step 1, 调用usdc合约approve()授权购买额度给platform合约，若成功，转step2；
Step 2, 调用platform合约trade()进行交易，若成功，转step 3；

Step 3, 调用transaction接口购买书籍。

书籍销毁流程（由作者主动发起）

Step 1, 调用platform合约burn(), 若成功, 转step 2;

Step 2, 调用issue接口保存销毁地址。

用户登录

登录流程:

Step 1, 前端连接metamask, 用户授权连接账户

Step 2, 连接成功后, 返回钱包地址到后端并获取nonce

Step 3, 前端使用nonce和钱包地址进行签名signature, 返回signature和address到后端

Step 4, 后端获取到signature和address, 通过address查询用户表获取nonce, 并使用nonce和address进行签名new_signature, 对比signature, 如果相同则登录成功, 否则登录失败



```
1 验证通过后,更新nonce,避免用户伪造登录
2 问题
3 {
4     1,当用户切换账户时,前端需要提醒用户是否重新登录
5     2,当用户切换网络时,前端提醒用户
6     3,当用户取消授权操作或签名操作时,需要清除登录状态
7     4,连接metamask钱包前需要验证用户是否安装了metamask
8 }
9 Pseudo Code
10 login {
11     if metamask not exists:
12         tell user('Metamask not found')
13     else:
14         connect to metamask;
```

```

15     if user approves:
16         get user address and current network;
17         call nonce api to get nonce;
18         ask user for signing with nonce;
19         if user signed:
20             sign with nonce and get signature;
21             call login api;
22             if login success:
23                 get token;
24                 tell user("login success");
25             else:
26                 clear login status;
27                 tell user("login fail");
28             else:
29                 clear login status;
30                 tell user("login fail");
31         else:
32             clear login status;
33             tell user("login fail");
34     }
35     listen metamask {
36         if user change account:
37             the metamask will warn user;
38         if user change network:
39             warn user("The network has been changed, please make sure that any act
40             ions you do on this network is what you want!");
41     }
42     API
43     {
44         1, login
45         url: http://{domain}/api/v1/login
46         data:
47         {
48             address: str,
49             signature: str
50         }
51         method: post
52         response:
53         201 {
54             token: str
55         }
56         or
57         400 {
58             detail: str
59         }
60         2, nonce
61         url: http://{domain}/api/v1/nonce
62         data:
63         {

```

```

63     address: str
64 }
65 method: POST
66 response:
67 201 {
68     nonce: str
69 }
70 or
71 400 {
72     detail: str
73 }
74 }

```

退出登录

权限校验：只有在登录状态可操作

操作流程：

Step 1，调用logout api，退出登录；

Step 2，前端清除登录状态，断开metamask；

```

1 api
2 {
3     1, logout
4     url: http://{domain}/api/v1/logout
5     method: post
6     response:
7     201 {
8         detail: str
9     }
10 }

```

首页

搜索接口

```

1 url: http://domain/api/v1/issues
2 method: get
3 params: {
4     search: str
5 }
6 response:
7 200 {
8     "count": 2,
9     "page_size": 20,
10    "page_count": 1,
11    "page": 1,
12    "next": null,
13    "previous": null,

```



```

26         n_circulations: '',
27         quantity: ''
28     }
29 },
30 ...
31 ]
32 }

```

用户详情浏览页(其他用户或者游客浏览)

用户信息

```

1 url: http://domain/api/v1/users/{id}
2 method: get
3 params: 无
4 response:
5     200 {
6         id: '',
7         name: 昵称,
8         address: 钱包地址,
9         desc: 描述,
10        avatar_url: 头像,
11        // 非作者返回空
12        statistic: {
13            total_volume: 预计总收益,
14            min_price: 最低价格,
15            max_price: 最高价格,
16            total_books: 已发布书籍数量,
17            sales: 销售额,
18            n_owners: 购买者数量,
19            n_destroyed: 销毁的书籍数量
20        }
21        website_url: 个人主页,
22        discord_url: discord主页,
23        twitter_url: twitter主页,
24        is_verified: 是否通过twitter认证,
25        is_fans: 是否是粉丝
26    }

```

个人藏品collection

```

1 url: http://domain/api/v1/assets
2 method: get
3 params: {
4     user: int, required
5 }
6 response:
7     200 {
8         "count": 2,

```



```

9     "page_size": 20,
10    "page_count": 1,
11    "page": 1,
12    "next": null,
13    "previous": null,
14    "results": [
15        {
16            id: '',
17            quantity: 收藏数量,
18            issue: {
19                id: '',
20                price: '',
21                n_circulations: '',
22                quantity: '',
23                published_at: '',
24                bookmark: {
25                    id: '',
26                    issue: issue id,
27                    current_page: 当前阅读页,
28                },
29                book: {
30                    id: '',
31                    author: {
32                        id: '',
33                        name: '',
34                        avatar_url: ''
35                    },
36                    title: '',
37                    cover_url: ''
38                }
39            }
40        },
41        ...
42    ]
43 }

```

交易活动记录activity

```

1 url: http://domain/api/v1/transactions
2 method: get
3 params: {
4     user: int, required
5 }
6 response:
7     200: {
8         "count": 2,
9         "page_size": 20,
10        "page_count": 1,
11        "page": 1,

```

```

12     "next": null,
13     "previous": null,
14     "results": [
15         {
16             id: '',
17             issue: {
18                 id: '',
19                 book: {
20                     id: '',
21                     title: '',
22                     cover_url: ''
23                 },
24                 ...
25             }
26             quantity: '',
27             price: '',
28             seller: {
29                 id: '',
30                 name: '',
31                 address: ''
32             },
33             buyer: {
34                 id: '',
35                 name: '',
36                 address: ''
37             },
38             created_at: ''
39         },
40         ...
41     ]
42 }

```

发布的书籍published book

```

1 url: http://domain/api/v1/issues
2 method: get
3 params: {
4     author: int required
5 }
6 response:
7     200 {
8         "count": 2,
9         "page_size": 20,
10        "page_count": 1,
11        "page": 1,
12        "next": null,
13        "previous": null,
14        "results": [
15            {

```

```

16         id: '',
17         book: {
18             id: '',
19             author: {
20                 id: '',
21                 name: '',
22                 avatar_url: ''
23             },
24             title: '',
25             cover_url: ''
26         },
27         price: '',
28         n_circulations: '',
29         quantity: '',
30         published_at: ''
31     },
32     ...
33 ]
34 }

```

个人详情页

用户接口

```

1 url: http://domain/api/v1/users/current
2 method: get
3 params: 无
4 response:
5     200 {
6         id: '',
7         name: '',
8         address: '',
9         desc: '',
10        avatar_url: '',
11        statistic: {
12            total_volume: 预计总收益,
13            min_price: 最低价格,
14            max_price: 最高价格,
15            total_books: 已发布书籍数量,
16            sales: 销售额,
17            n_owners: 购买者数量,
18            n_destroyed: 销毁的书籍数量
19        },
20        website_url: '',
21        discord_url: '',
22        twitter_url: '',
23        is_verified: ''
24    }

```

个人藏品collection

```
1 url: http://domain/api/v1/assets/current
2 method: get
3 params: {
4   user: int, required
5 }
6 response:
7   200 {
8     "count": 2,
9     "page_size": 20,
10    "page_count": 1,
11    "page": 1,
12    "next": null,
13    "previous": null,
14    "results": [
15      {
16        id: '',
17        quantity: '',
18        issue: {
19          id: '',
20          book: {
21            id: '',
22            title: '',
23            cover_url: ''
24          },
25          ...
26        }
27      },
28      ...
29    ]
30  }
```

已挂订单pending orders

```
1 url: http://domain/api/v1/trades/current
2 method: get
3 params: 无
4 response:
5   200 {
6     "count": 2,
7     "page_size": 20,
8     "page_count": 1,
9     "page": 1,
10    "next": null,
11    "previous": null,
12    "results": [
13      {
```

```

14         id: '',
15         issue: {
16             id: '',
17             book: {
18                 id: '',
19                 title: '',
20                 cover_url: ''
21             },
22             ...
23         }
24         price: '',
25         quantity: ''
26     },
27     ...
28 ]
29 }

```

交易活动记录activity

```

1 url: http://domain/api/v1/transactions/current
2 method: get
3 params: 无
4 response:
5     200: {
6         "count": 2,
7         "page_size": 20,
8         "page_count": 1,
9         "page": 1,
10        "next": null,
11        "previous": null,
12        "results": [
13            {
14                id: '',
15                issue: {
16                    id: '',
17                    book: {
18                        id: '',
19                        title: '',
20                        cover_url: ''
21                    },
22                    ...
23                }
24                quantity: '',
25                price: '',
26                seller: {
27                    id: '',
28                    name: '',
29                    address: ''
30                },

```

```
31     buyer: {
32         id: '',
33         name: '',
34         address: ''
35     },
36     created_at: ''
37 },
38 ...
39 ]
40 }
```

个人总收益

```
1 url: http://domain/api/v1/benefits/total
2 method: get
3 params: 无
4 response:
5     200: [
6         {
7             currency: 币种,
8             amount: 额度
9         },
10        ...
11    ]
```

草稿draft

```
1 url: http://domain/api/v1/drafts
2 method: get
3 params: 无
4 response:
5     200: {
6         "count": 2,
7         "page_size": 20,
8         "page_count": 1,
9         "page": 1,
10        "next": null,
11        "previous": null,
12        "results": [
13            {
14                id: '',
15                title: 标题,
16                content: 内容,
17                updated_at: 更新时间
18            },
19            ...
20        ]
21    }
```

即将上架to be listed

```
1 url: http://domain/api/v1/books
2 method: get
3 params: 无
4 response:
5   200: {
6     "count": 2,
7     "page_size": 20,
8     "page_count": 1,
9     "page": 1,
10    "next": null,
11    "previous": null,
12    "results": [
13      {
14        id: '',
15        title: 标题,
16        cover_url: 封面,
17        status: 上架状态
18      },
19      ...
20    ]
21  }
22 =====
23 edit按钮调用book api
24 publication按钮调用issue api
25 delete按钮调用book api
```

上架成功的书籍available

```
1 url: http://domain/api/v1/issues/current
2 method: get
3 params: 无
4 response:
5   200: {
6     "count": 2,
7     "page_size": 20,
8     "page_count": 1,
9     "page": 1,
10    "next": null,
11    "previous": null,
12    "results": [
13      {
14        id: '',
15        book: {
16          id: '',
17          title: '',
18          cover_url: '',
```

```

19         author: {
20             id: '',
21             avatar_url: '',
22             address: '',
23             name: ''
24         }
25     }
26     price: '',
27     quantity: '',
28     n_circulations: '',
29     status: ''
30 },
31 ...
32 ]
33 }

```

未售出的issue重新上架

```

1 url: http://domain/api/v1/issues/{id}/resale
2 method: patch
3 params: {
4     price: 发行价格, float
5     quantity: 发行数量, int
6     royalty: 版税, float, 0-100
7     buy_limit: 购买限制, int
8     published_at: 预售时间, str, datetime, 必须
9     duration: 销售时长, int, 单位分钟
10 }
11 response:
12     201: {
13         id: '',
14         book: {
15             id: '',
16             title: 书籍名称,
17             cover_url: 书籍封面,
18             desc: 书籍描述,
19             author: {
20                 id: '',
21                 name: 用户昵称,
22                 avatar_url: 头像,
23                 address: 钱包地址,
24                 website_url: 个人网站,
25                 discord_url: discord主页,
26                 twitter_url: twitter主页,
27             },
28             preview: {
29                 file_url: 预览文件url
30             },

```



```

31     },
32     price: 出版价格,
33     quantity: 出版数量,
34     published_at: 出售开始时间,
35     duration: 出售时长,
36     status: 出版状态,
37     n_circulations: 已售数量,
38     destroyed_log: NFT销毁记录地址,
39     buy_limit: 购买限制,
40     trade: {
41         id: '',
42         quantity: 挂单数量,
43         price: 挂单价格
44     },
45     n_owners: 购买者者数量,
46     is_wished: 是否添加心愿单,
47     token: {
48         id: NFT id,
49         contract_address: NFT合约地址,
50         block_chain: 区块链网络,
51         standard: 合约标准,
52         currency: NFT购买token
53     }
54 }
55 or
56 400 {}

```

书籍上传页

当添加一本新书时，不管是pdf或者草稿，都会启动一个task去发起书籍上传动作。如果是草稿，则需要先对草稿进行处理，然后再加密上传。g

```

1  url: http://domain/api/v1/books
2  method: post
3  params: {
4      file: pdf/epub/txt, 小于200M
5      cover: jpg/png, 小于5M, 必须
6      title: 长度150, 必须
7      desc: 长度1500, 必须
8      draft: 草稿id
9  }
10 response:
11     201: {
12         id: '',
13         title: 标题,
14         desc: 描述,
15         cover_url: 封面url,
16         draft: draft id

```

```
17 }
18 or
19 400 {}
```

书籍出版页

```
1 url: http://domain/api/v1/issues
2 method: post
3 params: {
4   book: 书籍id, 必须
5   price: 发行价格, 默认0
6   quantity: 发行数量, 默认1
7   royalty: 版税, 默认0
8   buy_limit: 购买限制, 默认1
9   published_at: 预售时间, 必须
10  duration: 销售时长, 整型数字, 单位分钟
11  token: {
12    block_chain: polygon/bnb二选一
13    currency: USDC
14  }
15 }
16 response:
17 201: {
18   id: '',
19   book: {
20     id: '',
21     title: 书籍名称,
22     cover_url: 书籍封面,
23     desc: 书籍描述,
24     author: {
25       id: '',
26       name: 用户昵称,
27       avatar_url: 头像,
28       address: 钱包地址,
29       website_url: 个人网站,
30       discord_url: discord主页,
31       twitter_url: twitter主页,
32     },
33     preview: {
34       file_url: 预览文件url
35     },
36   },
37   price: 出版价格,
38   quantity: 出版数量,
39   published_at: 出售开始时间,
40   duration: 出售时长,
41   status: 出版状态,
42   n_circulations: 已售数量,
```

```
43     destroyed_log: NFT销毁记录地址,
44     buy_limit: 购买限制,
45     trade: {
46         id: '',
47         quantity: 挂单数量,
48         price: 挂单价格
49     },
50     n_owners: 购买者者数量,
51     is_wished: 是否添加心愿单,
52     token: {
53         id: NFT id,
54         contract_address: NFT合约地址,
55         block_chain: 区块链网络,
56         standard: 合约标准,
57         currency: NFT购买token
58     }
59 }
60 or
61 400 {}
```

书籍详情页

书籍信息

```
1 url: http://domain/api/v1/issues/{id}
2 method: get
3 params: 无
4 response:
5     200: {
6         id: '',
7         book: {
8             id: '',
9             title: 书籍名称,
10            cover_url: 书籍封面,
11            desc: 书籍描述,
12            author: {
13                id: '',
14                name: 用户昵称,
15                avatar_url: 头像,
16                address: 钱包地址,
17                website_url: 个人网站,
18                discord_url: discord主页,
19                twitter_url: twitter主页,
20            },
21            preview: {
22                file_url: 预览文件url
23            },
24        },
```

```

25     price: 出版价格,
26     quantity: 出版数量,
27     published_at: 出售开始时间,
28     duration: 出售时长,
29     status: 出版状态,
30     n_circulations: 已售数量,
31     destroyed_log: NFT销毁记录地址,
32     buy_limit: 购买限制,
33     trade: {
34         id: '',
35         quantity: 挂单数量,
36         price: 挂单价格
37     },
38     price_range: {
39         min_price: 所有挂单中的最低价格,
40         max_price: 所有挂单中的最高价格
41     },
42     n_owned: 当前登陆用户拥有的数量,
43     n_owners: 购买者数量,
44     is_wished: 是否添加心愿单,
45     token: {
46         id: NFT id,
47         contract_address: NFT合约地址,
48         block_chain: 区块链网络,
49         standard: 合约标准,
50         currency: NFT购买token
51     }
52 }

```

二手市场挂单列表

```

1 url: http://domain/api/v1/trades
2 method: get
3 params: {
4     issue: id
5 }
6 response:
7     200: {
8         "count": 2,
9         "page_size": 20,
10        "page_count": 1,
11        "page": 1,
12        "next": null,
13        "previous": null,
14        "results": [
15            {
16                id: '',
17                user: {
18                    id: '',

```

```

19         name: '',
20         address: '',
21         avatar_url: ''
22     },
23     issue: {
24         id: '',
25         book: {
26             id: '',
27             ...
28         },
29         ...
30     },
31     quantity: 挂单数量,
32     price: 挂单价格,
33     first_release: 首发,
34     created_at: ''
35 },
36 ...
37 ]
38 }

```

二手市场交易列表activity

```

1 url: http://domain/api/v1/transactions
2 method: get
3 params: {
4     issue: id
5 }
6 response:
7     200: {
8         "count": 2,
9         "page_size": 20,
10        "page_count": 1,
11        "page": 1,
12        "next": null,
13        "previous": null,
14        "results": [
15            {
16                id: '',
17                issue: {
18                    id: '',
19                    book: {
20                        id: '',
21                        title: '',
22                        cover_url: '',
23                    },
24                    ...
25                },
26                quantity: 购买数量,

```

```

27     price: 购买单价,
28     seller: {
29         id: '',
30         address: '',
31         name: ''
32     },
33     buyer: {
34         id: '',
35         address: '',
36         name: ''
37     },
38     source: 1 or 2, # 1表示一级市场, 2表示二级市场
39     hash: 链上交易hash值,
40     created_at: ''
41 }
42 ]
43 }

```

二手市场交易趋势图trend

```

1 url: http://domain/api/v1/transactions/trend
2 method: get
3 params: {
4     issue: id
5 }
6 response:
7     200: {
8         dates: 日期列表,
9         quantities: 数量列表
10    }

```

用户信息修改页

信息修改

```

1 url: http://domain/api/v1/users/{id}
2 method: put/patch
3 params: {
4     avatar: jpg/png, 小于5M
5     banner: jpg/png, 小于5M
6     name: 长度150
7     desc: 长度1500
8     website_url: url
9     discord_url: url
10 }
11 response:
12     200 {
13         id: '',

```

```

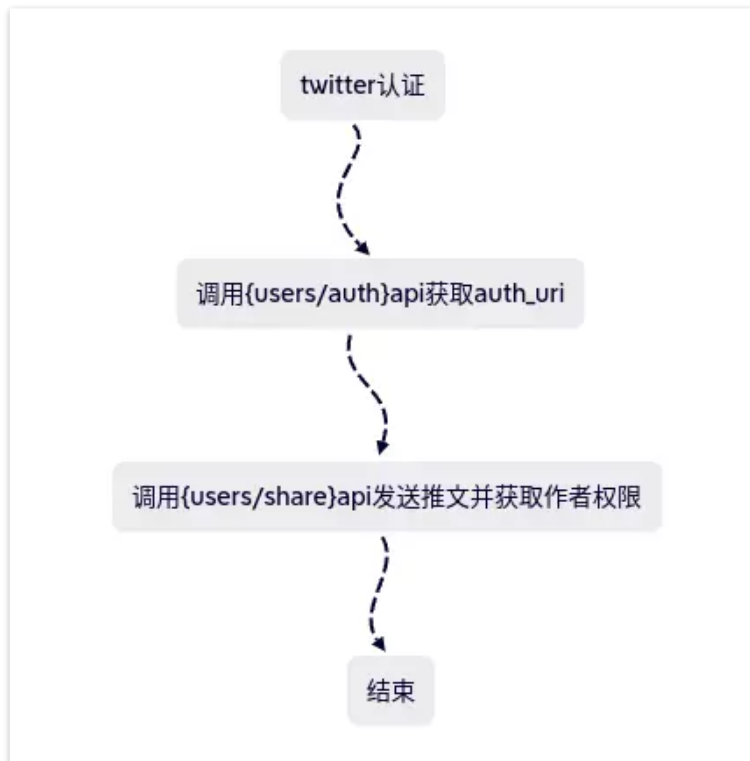
14     name: '',
15     address: '',
16     desc: '',
17     avatar_url: '',
18     // 非作者返回空
19     statistic: {
20         total_volume: '',
21         min_price: '',
22         max_price: '',
23         total_books: '',
24         sales: '',
25         n_owners: '',
26     }
27     website_url: '',
28     discord_url: '',
29     twitter_url: '',
30     is_verified: true
31 }

```

twitter认证

需要绑定用户的twitter账号和平台的账号，当对用户授权时可以直接通过关联查询获取到用户的平台账号并调用合同直接给用户授权。

身份验证：用户授权 -> 跳转链接 -> 发送推文 -> 授权用户 -> 返回主页



```

1 1, 获取用户twitter授权
2 url: http://domain/api/v1/users/auth
3 method: put/patch
4 params: 无
5 response:
6 200 {

```

```

7     auth_url: ''
8 }
9 2, 发送twitter并授权
10 url: http://domain/api/v1/users/share
11 method: put/patch
12 params: {
13     oauth_token: str, required, 来自auth_url
14     oauth_verifier: str, required, 来自auth_url
15     content: str, required
16 }
17 response:
18     200 {
19         'detail': 'success'
20     }

```

个人关注页

书籍收藏

```

1 添加收藏
2 url: http://domain/api/v1/wishlists
3 method: post
4 params: {
5     issue: 出版id
6 }
7 response:
8     200 {
9
10 }
11
12 收藏列表
13 url: http://domain/api/v1/wishlists/current
14 method: get
15 params: 无
16 response:
17     200 {
18         "count": 2,
19         "page_size": 20,
20         "page_count": 1,
21         "page": 1,
22         "next": null,
23         "previous": null,
24         "results": [
25             {
26                 id: '',
27                 issue: {
28                     id: '',
29                     quantity: '',

```



```

30     price: '',
31     n_circulations: '',
32     published_at: '',
33     book: {
34         id: '',
35         title: '',
36         desc: '',
37         cover_url: '',
38         author: {
39             id: '',
40             name: '',
41             avatar_url: '',
42             address: ''
43         }
44     },
45     token: {
46         id: '',
47         contract_address: '',
48         block_chain: '',
49         standard: '',
50         currency: ''
51     }
52 }
53 },
54 ...
55 ]
56 }

```

```

58 取消收藏
59 url: http://domain/api/v1/wishlists/remove
60 method: post
61 params: {
62     issue: 出版id
63 }
64 response:
65     204

```

关注的作者

```

1  添加关注
2  url: http://domain/api/v1/fans
3  method: post
4  params: {
5      author: 作者id
6  }
7  response:
8      200 {
9

```

```
10     }
11
12 关注列表
13 url: http://domain/api/v1/fans/current
14 method: get
15 params: 无
16 response:
17     200 {
18         "count": 2,
19         "page_size": 20,
20         "page_count": 1,
21         "page": 1,
22         "next": null,
23         "previous": null,
24         "results": [
25             {
26                 id: '',
27                 author: {
28                     id: '',
29                     avatar_url: '',
30                     name: '',
31                     address: ''
32                 }
33             },
34             ...
35         ]
36     }
37
38 取消关注
39 url: http://domain/api/v1/fans/remove
40 method: post
41 params: {
42     author: 作者id
43 }
44 response:
45     204
```

阅读

```
1 url: http://domain/api/v1/assets/{id}/read
2 method: get
3 params: 无
4 response:
5     200 {
6         file_url: 文件url,
7         bookmark: {
8             id: '',
9             current_page: 当前阅读页,
```

```
10     }
11 }
```

草稿

```
1  添加
2  url: http://domain/api/v1/drafts
3  method: post
4  params: {
5      title: 长度150, 必须
6      content: 长度1000000, 必须
7  }
8  response:
9      201 {
10         id: '',
11         title: '',
12         content: ''
13     }
14
15 修改
16 url: http://domain/api/v1/drafts/{id}
17 method: put/patch
18 params: {
19     title: 长度150
20     content: 长度1000000
21 }
22 response:
23     200 {
24         id: '',
25         title: '',
26         content: ''
27     }
28
29 删除
30 url: http://domain/api/v1/drafts/{id}
31 method: delete
32 params: 无
33 response:
34     204
```

书籍

```
1  修改
2  url: http://domain/api/v1/books/{id}
3  method: put/patch
4  params: {
```

```
5   file: pdf/epub/txt, 小于200M
6   cover: jpg/png, 小于5M
7   title: 长度150
8   desc: 长度1500
9   draft: 草稿id
10  }
11  response:
12    201: {
13      id: '',
14      title: 标题,
15      desc: 描述,
16      cover_url: 封面,
17      draft: 草稿id
18    }
19    or
20    400 {
21      title: [error message],
22      desc: ...,
23      ...
24    }
25    or
26    {
27      non_field_errors: []
28    }
29    or
30    {
31      detail: ''
32    }
33  删除
34  url: http://domain/api/v1/books/{id}
35  method: delete
36  params: 无
37  response:
38    204
```

出版

```
1  修改
2  url: http://domain/api/v1/issues/{id}
3  method: put/patch
4  params: {
5    price: 发行价格, 默认0
6    quantity: 发行数量, 默认1
7    royalty: 版税, 默认0
8    buy_limit: 购买限制, 默认1
9    published_at: 预售时间, 必须
10   duration: 销售时长, 整型数字, 单位分钟
11   token: {
12     block_chain: polygon/bnb二选一
```

```
13     currency: 默认USDC
14   }
15 }
16 response:
17   201: {
18     id: '',
19     book: {
20       id: '',
21       title: '',
22       cover_url: '',
23       desc: ''
24       author: {
25         id: '',
26         name: '',
27         avatar_url: '',
28         address: '',
29         website_url: '',
30         discord_url: '',
31       },
32       preview: {
33         file_url: ''
34       }
35     },
36     price: '',
37     quantity: '',
38     published_at: '',
39     duration: '',
40     status: '',
41     n_destroyed: '',
42     destroyed_log: '',
43     buy_limit: '',
44     trade: {
45       id: '',
46       quantity: '',
47       price: ''
48     },
49     n_owners: '',
50     token: {
51       id: '',
52       contract_address: '',
53       block_chain: '',
54       standard: '',
55       currency: ''
56     }
57   }
58   or
59   400 {}
```

61 删除

```
62 url: http://domain/api/v1/issues/{id}
63 method: delete
64 params: 无
65 response:
66     204
```

购买

```
1 url: http://domain/api/v1/transactions
2 method: post
3 params: {
4     trade: int, required
5     quantity: int, required
6     status: str, //一级市场不需要，二级市场根据链上交易状态定
7     hash: str //一级市场不需要，二级市场需要
8 }
9 response:
10     201 {
11         id: '',
12         issue: {
13             id: '',
14             book: {
15                 id: '',
16                 title: '',
17                 cover_url: ''
18             },
19             ...
20         },
21         quantity: '',
22         price: '',
23         seller: {
24             id: '',
25             name: '',
26             address: ''
27         },
28         buyer: {
29             id: '',
30             name: '',
31             address: ''
32         },
33         created_at: ''
34     }
```