用户模块

U1.新建用户（后续后台管理页面，管理员调用）

[POST] /v1/users

Request Body

```
{
    "work_no" : "7788", // 员工工号
    "username" : "hello", // 用户名
    "password" : "mypassword", // 密码
    "email" : "hello@gmail.com" // 邮箱
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

U2.用户登录

[POST] /v1/users/login

Request Body

```
{
    "work_no": "7856", // 员工工号
    "password": "mypassword" // 用户密码
}
```

Response Body

```
{
  "statusCode": 200,
  "msg": "success",
  "data": null
}
```

U3.获取用户元数据（用户登录系统时调用）

[GET] /v1/users/{current}/

- current: 查看当前用户的个人信息，可替换为user_id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "work_no": "7788", // 员工工号
        "username": "hello", // 用户名
        "email": "hello@Gmail.com" // 邮箱
    }
}
```

U4.注销登录

[POST] /v1/users/logout

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

U5.获取用户所在群组

[GET] /v1/users/{current}/groups

- current: 查看当前用户所在的群组，可替换为任意user_id，查看不同用户所在的群组

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": [
        {
            "group_id": "b57a392d-6510-4116-99db-f76cc16a78e5", // 群组id
            "group_name": "programmer", // 群组名称
            "creator_id": "1cbbf901-24ad-40fd-a35a-5dce15c82333", // 创建者id
            "created_at": "2019-02-23 07:30:25" // 创建时间
        },
        {
            "group_id": "c96c2465-62b0-47d6-b164-d51cb849deab",
            "group_name": "tester",
            "creator_id": "e1f5f562-2e96-4b3e-a6ff-e3f953c5b368",
            "created_at": "2019-06-16 12:45:12"
        }
    ]
}
```

群组模块

G1.新建群组

[POST] /v1/groups/

Request Body

```
{
    "group_name": "programmer" // 群组名称
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "group_id": "e3a10377-edcc-4a8a-8cce-396b0e223f48", // 群组id
        "group_name": "programmer", // 群组名称
        "creator_id": " 1cbbf901-24ad-40fd-a35a-5dce15c82333", // 创建者id
        "created_at": "2019-07-01 20:58:10" // 创建时间
    }
}
```

G2.获取群组元数据

[GET] /v1/groups/{group_id}

- group_id: 群组id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "group_id": "e3a10377-edcc-4a8a-8cce-396b0e223f48", // 群组id
        "group_name": "programmer", // 群组名称
        "creator_id": " 1cbbf901-24ad-40fd-a35a-5dce15c82333", // 创建者id
        "created_at": "2019-07-01 20:58:10" // 创建时间
    }
}
```

G3.修改群组元数据

[PUT] /v1/groups/{group_id}

- group_id: 群组id

Request Body

```
{
    "group_name": "newbility" // 群组名称
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

G4.删除群组

[DELETE] /v1/groups/{group_id}

- group_id: 群组id

Resposne Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

G5.添加群组用户

[POST] /v1/groups/{group_id}/members

- group_id: 群组id

Request Body

```
{
        // 用户id列表
    "usersIdList": [
        "1cbbf901-24ad-40fd-a35a-5dce15c82333",
        "24c16cd4-a2e6-4bd5-91b2-ab51c87b7514",
        "6272b618-a4e5-49c3-a4cb-69a94605c692"
    ]
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

G6.获取群组用户

[GET] /v1/groups/{group_id}/members

- group_id: 群组id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": [
        {
                "user_id": "1cbbf901-24ad-40fd-a35a-5dce15c82333", // 用户id
            "username": "jennifer", // 用户名
            "work_no": "7788" // 用户工号
        },
        {
            "user_id": "24c16cd4-a2e6-4bd5-91b2-ab51c87b7514",
            "username": "oliver",
            "work_no": "9527"
        }
    ]
}
```

G7.删除群组用户

[DELETE] /v1/groups/{group_id}/members/{member_id}

- group_id: 群组id
- member_id: 成员id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

检索模块

S1.搜索建议

[GET] /search/suggestions{?type,keyword,size}

```
GET /search/suggestions?type=all&keyword=算法&size=10
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": [
    "算法",
    "优化算法",
    "算法&数学",
    "算法分析",
    "算法导论",
    "算法设计",
    "算法初级",
    "随机算法",
    "算法&数据结构",
    "算法/数据结构"
  ]
}
```

S2.获取高度相关的类目标签

[GET] `/search/top-associations{?keyword,tag_count,category_count}`

```
GET /search/top-associations?keyword=算法&tag_count=5&category_count=5
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": {
    "tags": [
      {
        "id": 137,
        "title": "算法"
      },
      {
        "id": 2998,
        "title": "计算机"
      },
      {
        "id": 2697,
        "title": "计算机科学"
      },
      {
        "id": 66,
        "title": "哲学"
      },
      {
        "id": 133,
        "title": "编程"
      }
    ],
    "categories": [
      {
        "id": 6,
        "title": "科技"
      },
      {
        "id": 1,
        "title": "文学"
      },
      {
        "id": 2,
        "title": "流行"
      },
      {
        "id": 3,
        "title": "文化"
      },
      {
        "id": 4,
        "title": "生活"
      },
      {
        "id": 5,
        "title": "经管"
      }
    ]
  }
}
```

S3.搜索结果

[POST] `/search/results`

```
{
  "type": "all",
  "keyword": "算法",
  "tags": [
    1,
    3
  ],
  "categories": [
    1,
    2
  ],
  "exts": [
    "jpg",
    "all",
    "jpg",
    "gif",
    "doc",
    "pdf"
  ],
  "time_zone": "+8",
  "page": 2,
  "per_page": 40
}
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": {
    "group_by_created_time": [
      {
        "key": "全部",
        "doc_count": 33
      },
      {
        "key": "三天内",
        "doc_count": 6
      },
      {
        "key": "一周内",
        "doc_count": 15
      },
      {
        "key": "一个月内",
        "doc_count": 23
      },
      {
        "key": "三个月内",
        "doc_count": 25
      },
      {
        "key": "半年内",
        "doc_count": 27
      },
      {
        "key": "一年内",
        "doc_count": 27
      },
      {
        "key": "一年前",
        "doc_count": 0
      }
    ],
    "group_by_modified_time": [
      {
        "key": "全部",
        "doc_count": 33
      },
      {
        "key": "三天内",
        "doc_count": 6
      },
      {
        "key": "一周内",
        "doc_count": 15
      },
      {
        "key": "一个月内",
        "doc_count": 23
      },
      {
        "key": "三个月内",
        "doc_count": 25
      },
      {
        "key": "半年内",
        "doc_count": 27
      },
      {
        "key": "一年内",
        "doc_count": 27
      },
      {
        "key": "一年前",
        "doc_count": 0
      }
    ],
    "result": [
      {
        "id": "image_10432347",
        "title": "算法",
        "desc": "《算法(英文版•第4版)》作为算法领域经典的参考书，全面介绍了关于算法和数据结构的必备知识，并特别针对排序、搜索、图处理和字符串处理进行了论述。第4版具体给出了每位程序员应知应会的50个算法，提供了实际代码，而且这些Java代码实现采用了模块化的编程风格，读者可以方便地加以改造。本书配套网站提供了本书内容的摘要及更多的代码实现、测试数据、练习、教学课件等资源。《算法(英文版•第4版)》适合用作大学教材或从业者的参考书。",
        "type": "image",
        "ext": "jpg",
        "categories": [
          0,
          1,
          6
        ],
        "tags": [
          6,
          133,
          137,
          2552,
          2697,
          2998,
          22409,
          24310
        ],
        "creator": "green",
        "store_key": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
        "thumbnail": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
        "derived_files": [],
        "created_time": "2017-07-01 21:34:16",
        "modified_time": "2017-07-06 21:34:16",
        "version": 0,
        "original_id": "10432347",
        "parent_id": null
      }
    ]
  }
}
```

S4.搜索类目或标签

[GET] `/search/tags?{keyword, size}`

[GET] `/search/categories?{keyword, size}`

```
GET /search/tags?keyword=算法&size=5
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": [
      {
        "id": 6,
        "title": "科技",
                "desc": "......"
      },
      {
        "id": 1,
        "title": "文学",
        "desc": "......"
      },
      {
        "id": 2,
        "title": "流行"
      },
      {
        "id": 3,
        "title": "文化"
      },
      {
        "id": 4,
        "title": "生活"
      },
      {
        "id": 5,
        "title": "经管"
      }
        ]
}
```

目录文档模块

D1.新建资源

[POST] /v1/resources/

Request Body

```
{
        "cur_id": "e911f136-35ad-416a-b195-7b1fad4bd7f1 ", // 当前所在目录id
    "type": "dir" // 资源类型
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "resource_id": "b6519605-6132-4ba5-9039-cec0f7fc9fe3", // 资源id
        "resource_name": "undefined", // 默认资源名称
        "type": "dir", // 资源类型
        "creator_id": "5c397e61-ee45-4af1-b094-4363b5fdf305", // 创建者id
        "created_at": "2019-07-01 19:51:08" // 创建时间
    }
}
```

D2.获取资源元数据（业务数据库 PostgreSQL）

[GET] /v1/resources/{resource_id}

- resource_id: 资源id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "resource_id": "b6519605-6132-4ba5-9039-cec0f7fc9fe3", // 资源id
        "resource_name": "meeting", // 资源名称
        "type": "dir", // 资源类型
        "creator_id": "5c397e61-ee45-4af1-b094-4363b5fdf305", // 创建者id
        "created_at": "2019-07-01 19:51:08" // 创建时间
    }
}
```

D3.修改资源元数据

[PUT] /v1/resources/{resource_id}

- resource_id: 资源id

Request Body

```
{
    "resource_name": "meeting" // 资源名称
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

D4.删除资源

[DELETE] /v1/resources/{resource_id}

- resource_id: 资源id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

D5.获取下级目录或挂载文档

[GET] /v1/resources/{resource_id}/slaves

- resource_id: 资源id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": [
        {
            "resource_id": "573d9b62-9e07-430c-b2a0-4825fbccc785", // 资源id
            "resource_name": "七月例会", // 资源名称
            "type": "dir", // 资源类型
            "creator": "小组长", // 创建者名称
            "created_at": "2019-07-01 09:21:28" // 创建时间
        },
        {
            "resource_id": "627f3add-e93a-435d-bd39-2f8023253f35",
            "resource_name": "八月例会",
            "type": "dir",
            "creator": "小组长",
            "created_at": "2019-08-01 09:30:21"
        }
    ]
}
```

D6.获取对该资源有操作权限的群组信息

[GET] /v1/resources/{resource_id}/authgroups

- resource_id: 资源id

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": [
        {
            "group_id": "b64b725b-ef13-4e3f-9d98-ddb3152981a6", // 群组id
            "group_name": "manager", // 群组名称
            "permission": "111" // 权限
        }
    ]
}
```

D7.获取文档meta

[GET] `/docs/{doc_id}`

```
GET /docs/1
```

Response Body

```
{
    "status": 200,
    "msg": "OK",
    "data": {
        "id": "1",
        "title": "code",
        "desc": "代码仓库",
        "creator": "green",
        "files": [
            "1",
            "2",
            "ABC"
        ],
        "meta_state": 1,
        "created_time": "2019-07-05 23:09:00",
        "modified_time": "2019-07-05 23:10:00"
    }
}
```

D8.更新文档meta

[PATCH] `/docs/{doc_id}`

```
PATCH /files/1
```

Request Body

```
{
    "title": "code",
    "desc": "代码仓库"
}
```

Response Body

```
{
    "status": 200,
    "msg": "OK",
    "data": {
        "id": "1",
        "title": "code",
        "desc": "代码仓库",
        "creator": "green",
        "files": [
            "1",
            "2",
            "ABC"
        ],
        "meta_state": 0,
        "created_time": "2019-07-05 23:09:00",
        "modified_time": "2019-07-05 23:10:00"
    }
}
```

文件模块

F1.获取policy（针对于使用阿里云 oss 实施的项目）

[POST] /v1/file/policy/${dir_uuid}

- ${dir_uuid} 为上传到逻辑文档的 uuid

Request Body:

```
{
    "ext" : "" // 准备上传文件的拓展名
}
```

Response Body:

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "accessKey": "", // 用户的 accessKey
        "callback": "", // 应用服务器的 /callback 接口
        "dir": "", // oss 的路径
        "expire": "", // policy 有效时间
        "host": "http://graduation-pro.oss-cn-hangzhou.aliyuncs.com", // oss 的域名
        "policy": "",
        "signature": "",
        "file_uuid": "" // 后面文件直传 oss 时需要的 filename
        "creator": "" //创建者id
    }
}
```

F2.获取签名URL（针对于使用Minio实施的项目）

[POST] /v1/file/url/${dir_uuid}

- ${dir_uuid} 为上传到逻辑文档的 uuid

Request Body:

```
{
    "ext" : "" // 准备上传文件的拓展名
}
```

Response Body:

```
{
    "statusCode": 200,
    "msg": "success",
    "data": {
        "url" : "" // 后续使用 PUT 方法上传文件
    }
}
```

F3. 文件直传 oss（针对阿里云 oss 实施）

[POST] **${host}**

- 此处的 host 是获取 /policy 接口返回 body 中的 host 字段

Request Body:

```
{
    "title": "文件名",
    "doc_id": "",  //当前上传文件所属的文档ID
    "parent_id": "",  //当前上传文件如果为某一文件的新版本，则需要传其父版本文件的ID，否则为""
    "store_key": "",  //文件在OSS中的key，由获取policy请求返回的 dir + / + 文件名组成 例：user-dir-
prefix/${filename}.${suffix}
```

```
    "creator": "", // 文件的创建者
    "size": "", // 文件的大小
    "filename": "", //存储在OSS里的文件名
    "policy": "",  //获取policy请求返回的policy字段
    "accessKey": "",  //获取policy请求返回的accessid字段
    "success_action_status": 200  //回调成功返回的状态码
    "callback": "",  //获取policy请求返回的callback字段
    "signature": "",  //获取policy请求返回的signature字段
    "file": (binary)  //所上传文件的二进制文件
}
```

Response Body:

```
{
    "Status": "OK"
}
```

F4. 文件直传 oss（针对 Minio 实施）

[PUT] **${host}**

- 此处的 host 是获取 A2 接口返回 body 中的 url 字段

Request Body:

```
{
        "file": (binary)  //所上传文件的二进制文件
}
```

F5. 上传文件成功后更新Meta（针对 Minio 实施）

[POST] /v1/file/meta

Request Body:

```
{
    "title": "", //"文件名"
    "store_key": "", //文件在Minio中的key。filename可从获取签名url请求返回的url获取 例: user-dir-
prefix/${filename}.${suffix}
    "doc_id": "", //当前上传文件所属的文档ID
    "parent_id": "", //当前上传文件如果为某一文件的新版本，则需要传其父版本文件的ID，否则为""
    "filename": "",
    "creator": "", // 文件的创建者
        "size": "" // 文件的大小
}
```

Response Body:

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

F6. 删除文件

[DELETE] /v1/file

Request Body:

```
{
        "dir_uuid": "", //文件所处的文档uuid
        "file_id":["","",""] //删除的文件id
}
```

Response Body:

```
{
        "statusCode": 200,
        "msg": "success",
        "data": null
}
```

F7. 下载文件

[GET] /v1/file/$dir_u uid/${file_id}

- ${dir_uuid}为下载的文件所在的文档的uuid
- ${file_id}为下载文件的id

Response Body: HttpServletResponse

F8. 获取文件历史版本列表

[GET] /v1/file/version/$dir_u uid/${file_id}

- ${dir_uuid}为下载的文件所在的文档的uuid
- ${file_id}为下载文件的id

Response Body:

```
{
        "statusCode": 200,
        "msg": "success",
        "data": {
                "file": [
            {
                "filename": "", //OSS里的文件名
                "title": "", //文件名
                "file_thumbnail": "", //文件缩略图（varchar）Base64编码的字符串
                "file_version": "", //文件版本号
                "parent_id": "", //父版本文件UUID
                "create_time": "", //文件创建时间
                "modified_time": "" //文件最后修改时间
            },
                        {
                ...
            }
                ]
        }
}
```

F9.Retrieve File Meta

获取文件meta

[GET] `/files/{file_id}`

```
GET /files/1
```

Response Body

```
{
    "id": "image_10432347",
    "title": "算法",
    "desc": "《算法(英文版•第4版)》作为算法领域经典的参考书，全面介绍了关于算法和数据结构的必备知识，并特别针对排序、搜索、图处理和字符串处理进行了论述。第4版具体给出了每位程序员应知应会的50个算法，提供了实际代码，而且这些Java代码实现采用了模块化的编程风格，读者可以方便地加以改造。本书配套网站提供了本书内容的摘要及更多的代码实现、测试数据、练习、教学课件等资源。《算法(英文版•第4版)》适合用作大学教材或从业者的参考书。",
    "creator": "green",
    "doc_id": "1",
    "type": "image",
    "ext": "jpg",
    "size": 1024,
    "categories": [
        0,
        1,
        6
    ],
    "tags": [
        6,
        133,
        137,
        2552,
        2697,
        2998,
        22409,
        24310
    ],
    "store_key": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
    "thumbnail": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
    "derived_files": [],
    "created_time": "2017-07-01 21:34:16",
    "modified_time": "2017-07-06 21:34:16",
    "version": 0,
    "original_id": "10432347",
    "parent_id": null
}
```

F10.Update File Meta

更新文件meta

> 部分更新，仅可更新部分字段

[PATCH] `/files/{file_id}`

```
PATCH /files/1
```

Request Body

```
{
    "title": "算法",
    "desc": "《算法(英文版•第4版)》作为算法领域经典的参考书，全面介绍了关于算法和数据结构的必备知识，并特别针对排序、搜索、图处理和字符串处理进行了论述。第4版具体给出了每位程序员应知应会的50个算法，提供了实际代码，而且这些Java代码实现采用了模块化的编程风格，读者可以方便地加以改造。本书配套网站提供了本书内容的摘要及更多的代码实现、测试数据、练习、教学课件等资源。《算法(英文版•第4版)》适合用作大学教材或从业者的参考书。",
    "categories": [
        0,
        1,
        6
    ],
    "tags": [
        6,
        133,
        137,
        2552,
        2697,
        2998,
        22409,
        24310
    ]
}
```

Response Body

```
{
    "status": 200,
    "msg": "OK",
    "data": {
        "id": "image_10432347",
        "title": "算法",
        "desc": "《算法(英文版•第4版)》作为算法领域经典的参考书，全面介绍了关于算法和数据结构的必备知识，并特别针对排序、搜索、图处理和字符串处理进行了论述。第4版具体给出了每位程序员应知应会的50个算法，提供了实际代码，而且这些Java代码实现采用了模块化的编程风格，读者可以方便地加以改造。本书配套网站提供了本书内容的摘要及更多的代
码实现、测试数据、练习、教学课件等资源。《算法(英文版•第4版)》适合用作大学教材或从业者的参考书。",
        "creator": "green",
        "doc_id": "1",
        "type": "image",
        "ext": "jpg",
        "size": 1024,
        "categories": [
            0,
            1,
            6
        ],
        "tags": [
            6,
            133,
            137,
            2552,
            2697,
            2998,
            22409,
            24310
        ],
        "store_key": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
        "thumbnail": "http://douban-test.oss-cn-beijing.aliyuncs.com/img/10432347.jpeg",
        "derived_files": [],
        "created_time": "2017-07-01 21:34:16",
        "modified_time": "2017-07-06 21:34:16",
        "version": 0,
        "original_id": "10432347",
        "parent_id": null
    }
}
```

F11.Create Tag / Category

[POST] `/tags/`

[POST] `/categories/`

Request Body

```
{
    "title": "算法",
    "desc": "......"
}
```

Response Body

```
{
    "status": 200,
    "msg": "OK",
    "data": {
        "id": 1,
        "title": "算法",
        "desc": "......"
    }
}
```

F12.Retrieve Tag / Category

[GET] `/tags/{tag_id}`

[GET] `/categories/{category_id}`

```
GET /tags/1
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": {
        "id": 1,
        "title": "算法",
        "desc": "......"
        }
}
```

F13. Update Tag / Category

[PUT] `/tags/{tag_id}`

[PUT] `/categories/{category_id}`

```
PUT /tags/1
```

Request Body

```
{
    "title": "算法",
    "desc": "......"
}
```

Response Body

```
{
    "status": 200,
    "msg": "OK",
    "data": {
        "id": 1,
        "title": "算法",
        "desc": "......"
    }
}
```

F14.Delete Tag / Category

[DELETE] `/tags/{tag_id}`

[DELETE] `/categories/{category_id}`

```
DELETE /tags/1
```

Response Body

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

F15.Get File Tags / Categories

[GET] `/files/{file_id}/tags`

[GET] `/files/{file_id}/categories`

```
GET /files/1/tags
```

Response Body

```
{
  "status": 200,
  "msg": "OK",
  "data": [
      {
        "id": 137,
        "title": "算法",
        "desc": "......"
      },
      {
        "id": 2998,
        "title": "计算机",
        "desc": "......"
      },
      {
        "id": 2697,
        "title": "计算机科学"
      },
      {
        "id": 66,
        "title": "哲学"
      },
      {
        "id": 133,
        "title": "编程"
      }
    ]
}
```

F16.Update File Tags / Categories

[PUT] `/files/{file_id}/tags`

[PUT] `/files/{file_id}/categories`

```
PUT /files/1/tags
```

Request Body

```
{
    "tags": [1, 2, 3]
}
```

Response Body

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

权限模块

F1.授予群组对指定目录或文档的操作权限

[POST] /v1/resources/{resource_id}/permissions

- resource_id: 资源id

Request Body

```
{
    "permission": "100", // 权限
    // 群组id列表
    "groupsIdList": [
        "b64b725b-ef13-4e3f-9d98-ddb3152981a6",
        "b57a392d-6510-4116-99db-f76cc16a78e5"
    ]
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```

F2.撤销群组对指定目录或文档的操作权限

[DELETE] /v1/resources/{resource_id}/permissions

- resource_id: 资源id

Request Body

```
{
    "group_id": "b57a392d-6510-4116-99db-f76cc16a78e5" // 群组id
}
```

Response Body

```
{
    "statusCode": 200,
    "msg": "success",
    "data": null
}
```