**Asset management api**

Mỗi respone từ server để sẽ gửi thêm mã status code và messages ở phần body, giúp kiểm soát lỗi ở client và điều chỉnh request. Duy nhất trường hợp các request có action là DELETE, khi thành công thì trả về trống

1. **Authentication**

**Đăng ký**

POST http://localhost:3000/api/auth/register

Content-Type: application/json

{

"password": "123456",

"fullName": "member13",

"studentCode": "0013",

"phoneNumber":"0013",

"email": "member13@gmail.com"

}

Thông tin đăng kí phải đầy đủ, không được trùng studentCode hoặc email

**Đăng nhập**

POST http://localhost:3000/api/auth/login

Content-Type: application/json

{

"password": "123456",

"email": "member1@gmail.com"

}

Sau khi đăng nhập, liên lạc giữa client và server sẽ được duy trì bởi sessionId lưu trong cookie của client.

Tài khoản chỉ đăng nhập được sau khi đăng ký và được enable bởi admin

**Đăng xuất**

POST <http://localhost:3000/api/auth/logout>

Vô hiệu hóa sessionId, không có quyền truy cập các resource có yêu cầu đăng nhập.

1. **Item**

**Resource:** <http://localhost:3000/api/item>

**Lấy tất cả các items để hiển thị ra main**

* Action: GET
* Không yêu cầu đăng nhập
* Nếu không có lỗi, server trả về array **items** ở phần body

**Lấy item bởi id**

* Action: GET
* Không yêu cầu đăng nhập
* Thêm id vào cuối resource
* Nếu không có lỗi, server trả về **item** ở phần body

**Thêm mới item**

* Action: POST
* Yêu cầu đăng nhập, có quyền admin
* Gửi json

{

"name" : "item10",

"quantity" : "99",

"category" : "aaaa",

"description" : "hello world"

}

* Name phải là unique, báo lỗi nếu trùng
* Nếu thêm mới thành công, trả về object **item**

**Cập nhật thông tin của item**

* Action: PATCH
* Yêu cầu đăng nhập, có quyền admin
* Thêm id vào cuối resource
* Phần body bao gồm nội dung cần cập nhật (json)
* Nếu thành công trả về **item** đã được cập nhật ở phần body

**Xóa item bằng id**

* Action: DELETE
* Yêu cầu đăng nhập, quyền admin
* Thêm id vào cuối resource
* Nếu không có lỗi gì (xóa thành công), sẽ không có gì được trả về

1. **User**

**Resource:** http://localhost:3000/api/user/

**Lấy tất cả người dùng**

* Action: GET
* Yêu cầu đăng nhập, quyền admin
* Nếu không có lỗi, trả về array **users** chứa tất cả người dùng ở phần body json

**Lấy thông tin người dùng bởi id**

* Action: GET
* Yêu cầu đăng nhập, quyền admin
* Thêm id của người dùng vào cuối resource
* Nếu không có lỗi, trả về **user** chứa thông tin người dùng cần tìm

**Đổi quyền truy cập của một user**

* Action: PATCH
* Yêu cầu đăng nhập, quyền admin
* Thêm id vào sau resource, thêm filter **togglePermission**
* Nếu **togglePermission=enable** thay đổi quyền đăng nhập của người dùng (giá trị enable bị đảo lại, quyền admin luôn false)
* Nếu **togglePermission=admin** thay đổi quyền admin (quyền admin bị đảo ngược, quyền đăng nhập luôn true)
* Trả về **user** chứa thông tin sau update của người dùng nếu không có lỗi

**Xóa người dùng bằng id**

* Ation: DELETE
* Yêu cầu đăng nhập, quyền admin
* Thêm id của người dùng vào cuối resource
* Nếu thành công, không trả về gì cả

1. **Order**

Resource: http://localhost:3000/api/order/

**Lấy tất cả đơn hàng**

* Action: GET
* Yêu cầu đăng nhập, quyền admin
* Nếu không có lỗi, trả về array **orders**, chưa tất cả đơn hàng ở phần body

**Lấy đơn hàng bằng id**

* Action: GET
* Yêu cầu đăng nhập, **cần fix lại**
* Thêm id của order vào cuối resource
* Nếu không có lỗi trả về **order** chứa thông tin đơn hàng

**Tạo mới đơn hàng**

* Action: POST
* Yêu cầu đăng nhập, không cần quyền admin
* Gửi json

{

"quantity": "9",

"idItem": "6290b766372ca2ed6ff00c8c",

"returnDate": "2022-12-30"

}

* Trong đó quantity là số lượng, idItem là id của vật dụng cần mượn, returnDate là ngày hẹn trả
* Khi tạo mới đơn hàng, trạng thái của nó là pending và chờ admin duyệt.

**Thay đổi trạng thái của đơn hàng**

* Action: PATCH
* Yêu cầu đăng nhập, quyền admin
* Thêm id vào sau resource, thêm filter **action**
* Nếu **action=deny** thì sẽ thay đổi trạng thái đơn hàng thành **denied** nếu ban đầu nó **pending**
* Nếu **action=accept** thì thay đổi trạng thái đơn thành **ok** nếu ban đầu là **pending. Accept** thành công chỉ khi số lượng sản phẩm sẵn có đủ để cho mượn
* Nếu **action=confirm** thì thay đổi trạng thái đơn hành thành done nếu trạng thái ban đầu là **ok**
* Nếu các thao tác hoàn tất, trả về order chứa thông tin đơn hàng

1. **Mô hình trạng thái đơn hàng**

