USER GUIDE

TÚ.HUỲNH - 15/02/2024 -

GIỚI THIỆU

Mockoon (Link download: https://mockoon.com/)

Là một công cụ API được sử dụng để tạo và quản lý các mock API. Nó cho phép các nhà phát triển tạo ra các API giả lập một cách nhanh chóng và dễ dàng để thử nghiệm, phát triển và kiểm thử ứng dụng mà không cần phải phụ thuộc vào các dịch vụ API thực tế. Công cụ này thường được sử dụng trong quá trình phát triển phần mềm để giả lập các phản hồi từ các API thực tế mà chưa được triển khai hoặc đang phát triển.

Lợi ích:

Mockoon mang lại lợi ích cho mọi vai trò trong phát triển ứng dụng:

- **Tiết kiệm thời gian:** Tạo và triển khai API nhanh chóng, tăng tốc độ phát triển.
- **Nâng cao chất lượng:** Phát hiện và sửa lỗi sớm, giảm thiểu rủi ro.
- **Tăng cường cộng tác:** Chia sẻ API giả lập dễ dàng, thúc đẩy hiệu quả làm việc.

Mục tiêu:

- Tạo API giả lập cho chức năng cụ thể
- Kiểm thử chức năng cụ thể

Công cụ:

- Mockoon
- Postman

ĐỐI TƯỢNG SỬ DỤNG

Ứng dụng của Mockoon có thể được phân thành ba khía cạnh chính

Nhân viên kiểm thử:

- **Kiểm tra mọi kịch bản:** Tạo API giả lập để kiểm tra ứng dụng trong mọi tình huống, không phụ thuộc vào dịch vụ thực tế.
- **Tạo phản hồi đa dạng:** Kiểm tra khả năng xử lý lỗi và các trường hợp bất ngờ của ứng dụng.

Nhân viên Triển khai:

- **Phát triển nhanh hơn:** Triển khai dịch vụ giả lập để tiết kiệm thời gian chờ dịch vụ thực tế.
- **Kiểm thử sớm:** Kiểm tra ứng dụng ngay từ giai đoạn đầu phát triển, phát hiện lỗi sớm hơn.

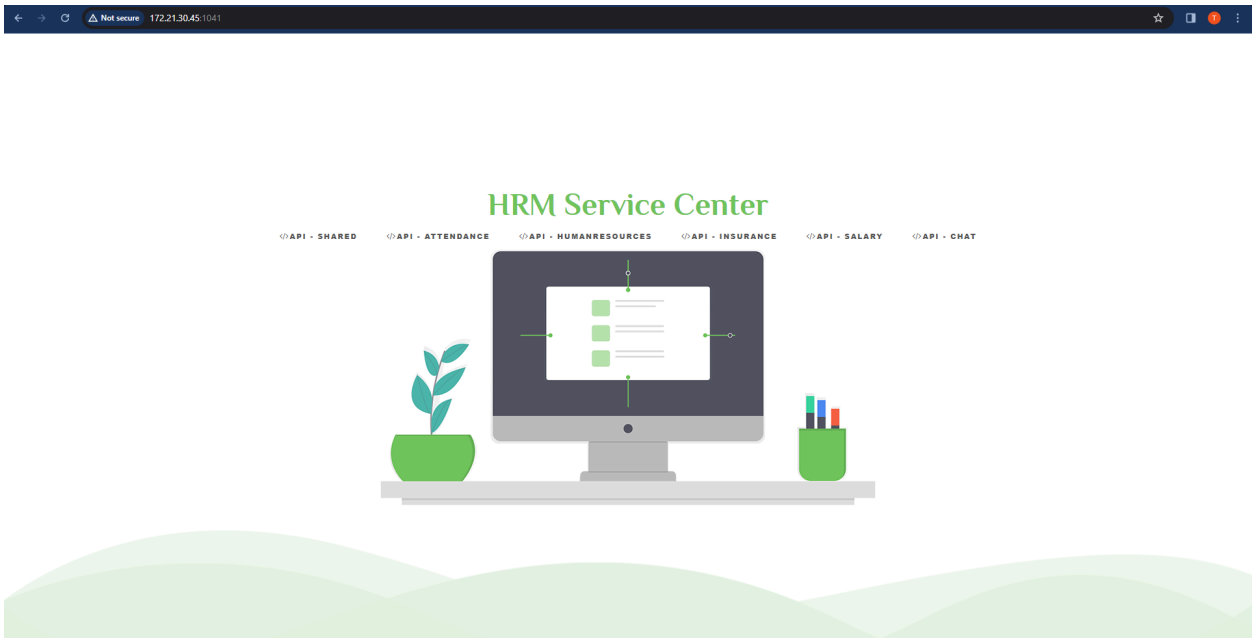
Nhân viên kỹ thuật:

- **Tích hợp vào quy trình phát triển:** Tạo API giả lập cho kiểm thử tự động và phát triển liên tục.
- **Giả lập dịch vụ thực tế:** Giúp kiểm thử ứng dụng khi dịch vụ thực tế không khả dụng.

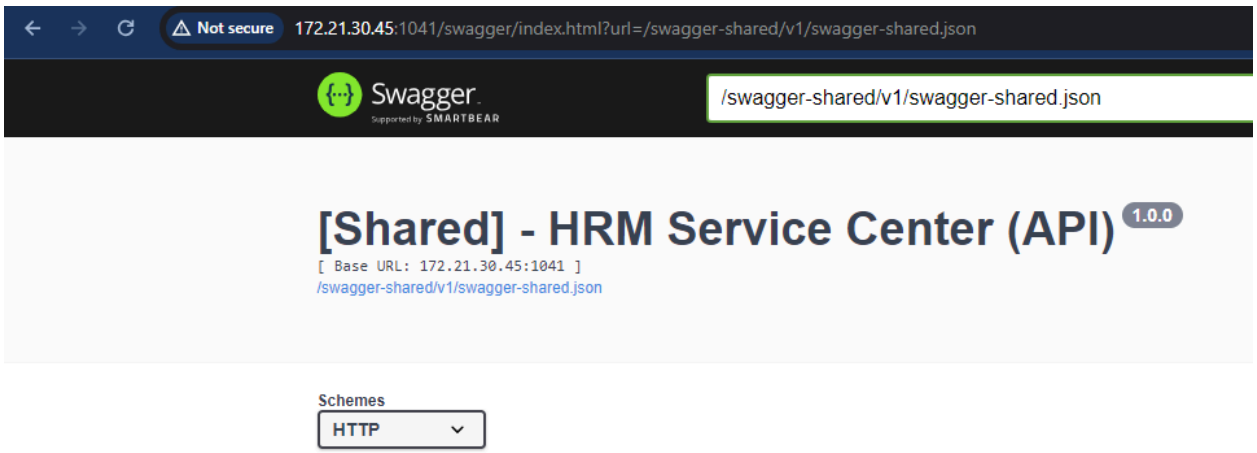
HƯỚNG DẪN CHI TIẾT

1. Import môi trường từ file Json

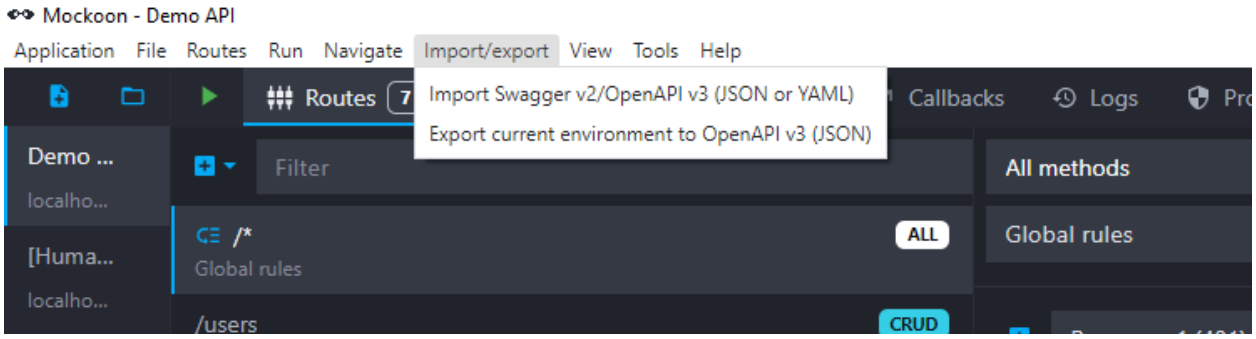
- Vào trang api swagger api v3. Chọn module.



- Nhấn vào link file .json mở file trên trình duyệt → copy nội dung lưu vào 1 file định dạng json (lưu tên bất kỳ đều được).



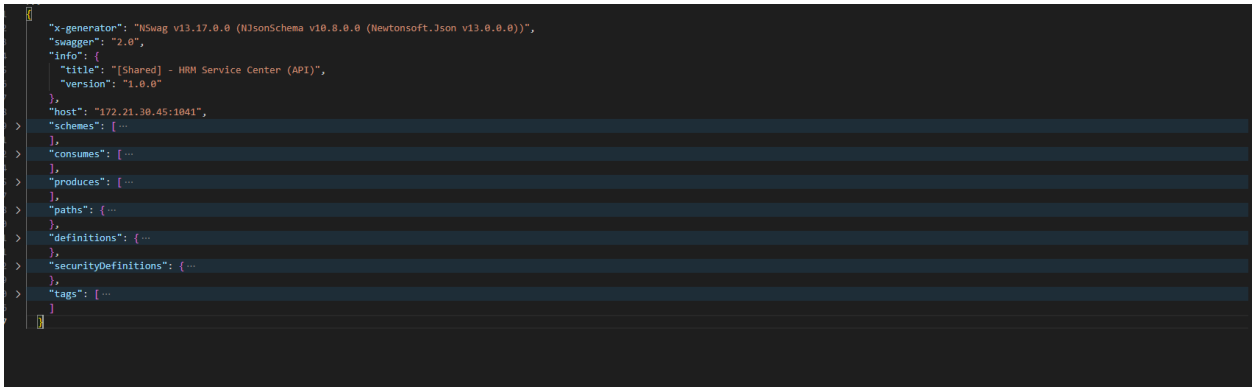
- Vào mockoon, chọn import/export → chọn Import Swagger v2/OpenAPI v3 (JSON or YAML) → chọn file json vừa mới tạo.



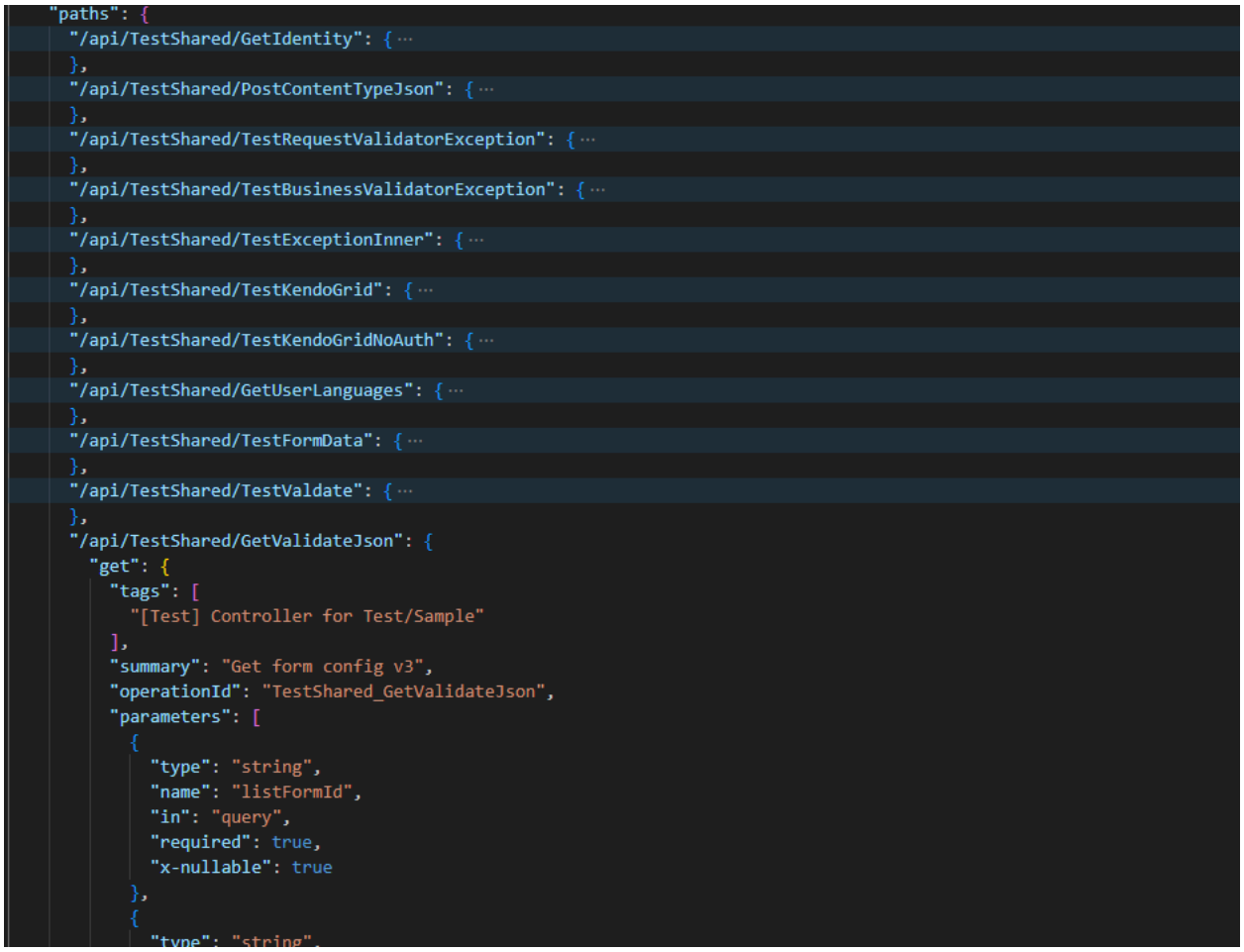
- Chọn file json tải về từ swagger → mockoon sẽ thiết lập môi trường dựa trên file swagger → chọn đường dẫn lưu file json chứa môi trường mockoon.

2. Edit file lấy từ Swagger để chỉ lấy những API nào cần dùng.

a. Cấu trúc file json lấy được từ trang Swagger:



b. Xóa hết nhưng đường dẫn api không cần dùng trong paths



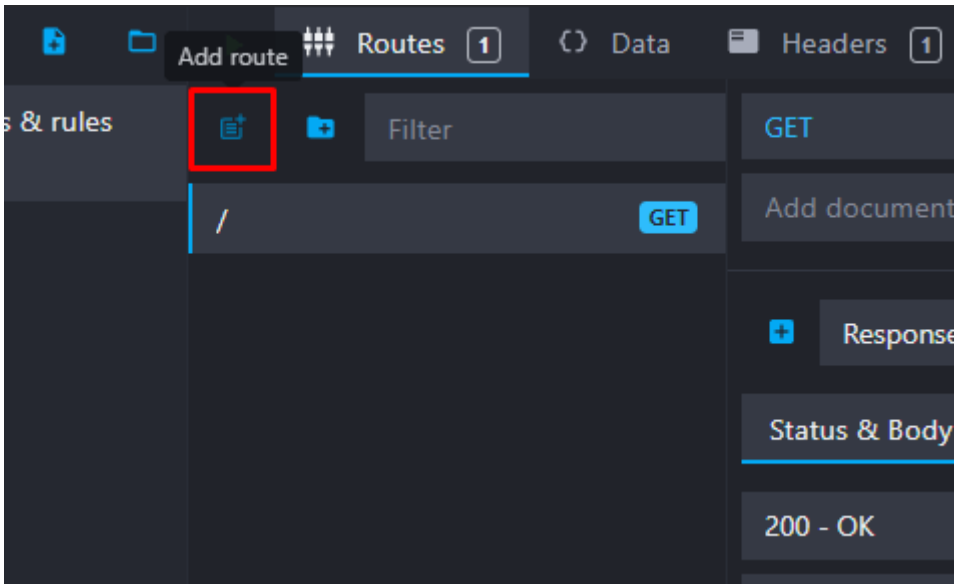
3. Cấu hình API

Mỗi response sẽ tương ứng với 1 test case.

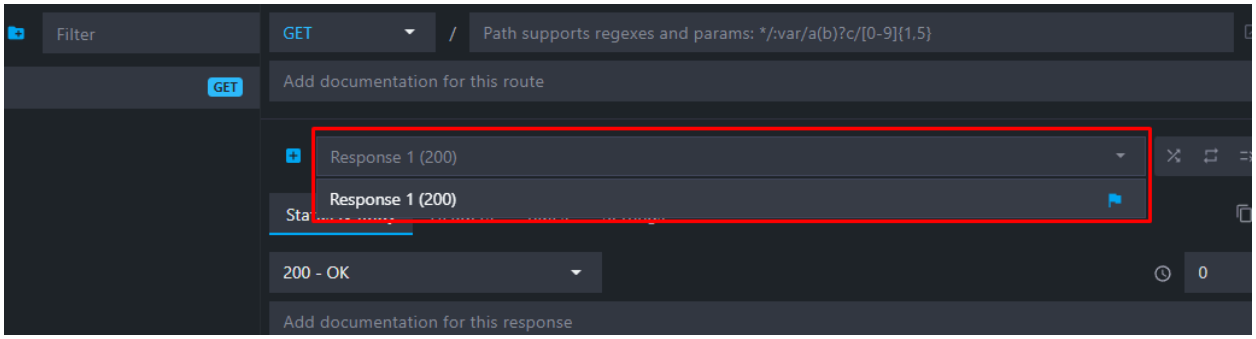
TestCase ID	TestCase Name	TestCase Description (Objective)	Pre-Condition	Method	API URL (Adjustable)	Request Headers	Request Body	Expected Response Status Code	Expected Message	Expected Headers	Expected Body
TC-10	Thông báo lỗi khi nhập vượt quá độ dài ký tự cho phép của trường EmpCode		Có access token	POST	To be update	Authorization: Bearer {AccessToken}	{ dt:[{ "EmpCode": [1 đoạn văn bản hơn 8000 ký tự] "EmpFullName": "Lê Minh Quốc", "Amount": "5000000", "EffectDate": "2023/07/23", "ReasonType": }, {},] }	200			{ "code": "-101", "message": "Invalid format required field" }

Step 1: Tạo route mới

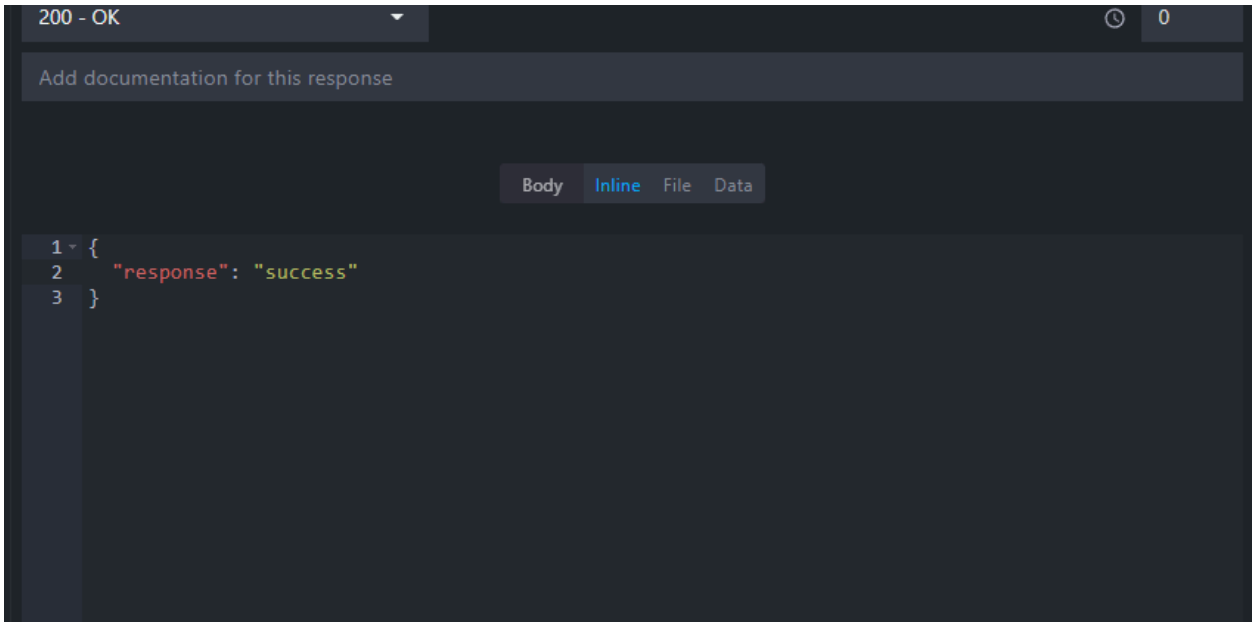
Nhấn button add route để thêm API



Khi tạo mới 1 API, sẽ luôn có ít nhất 1 response mặc định với mã lỗi 200.



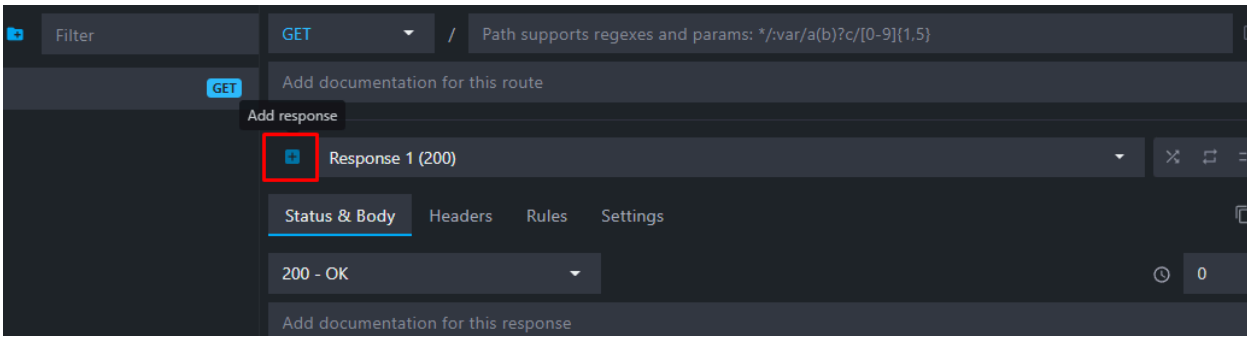
Cấu hình body trả về cho response.



Step 2. Thêm response "401 Unauthorized"

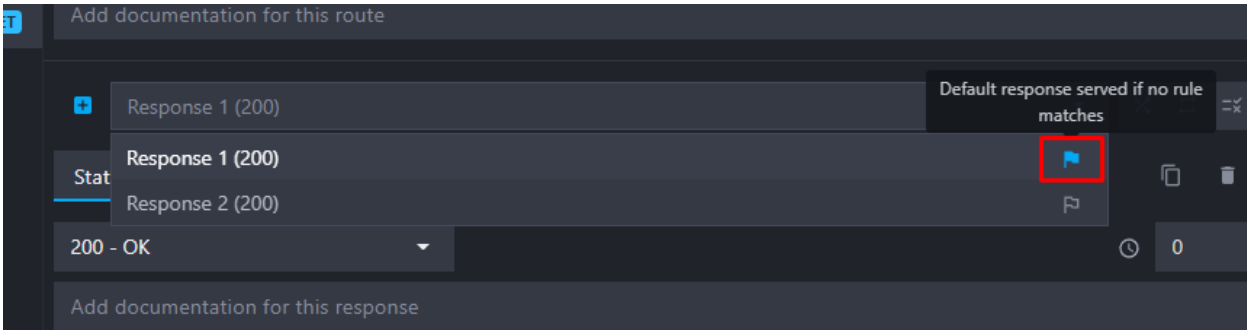
Giả sử trường hợp mong muốn trả về **401** khi request thiếu **headers Authorization**

Nhấn vào **plus** button ở vùng response để thêm mới response.

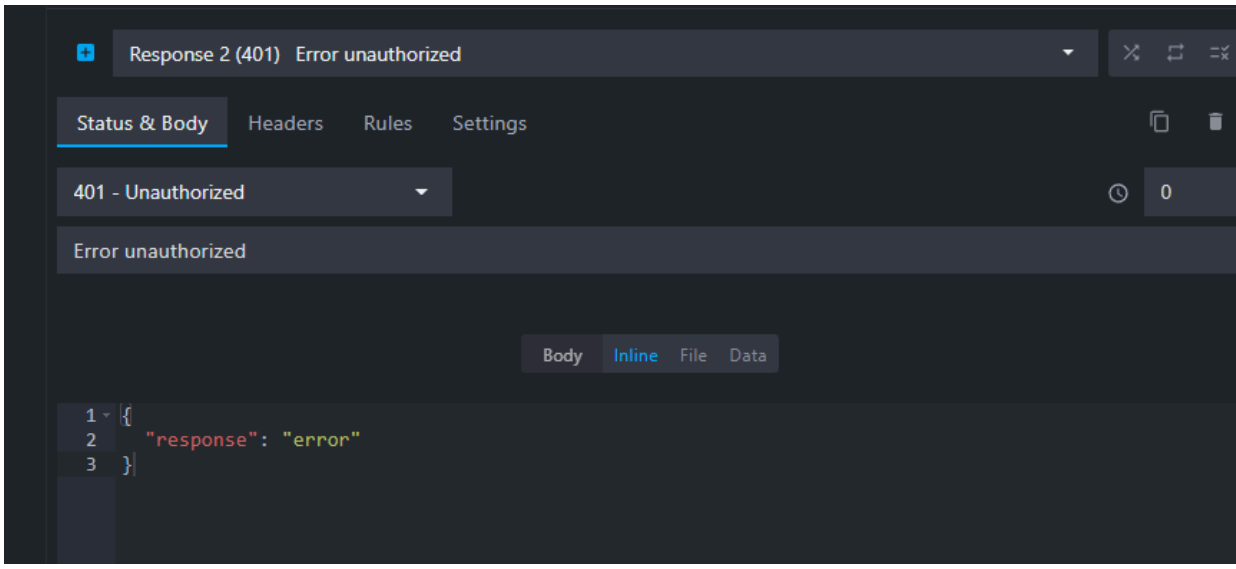


Có thể duplicate response để cấu hình cho những case tương tự.

Khi response nào được check flag xanh sẽ được mặc định trả về khi request tới không khớp với rules của response nào hoặc khi không cấu hình rules.



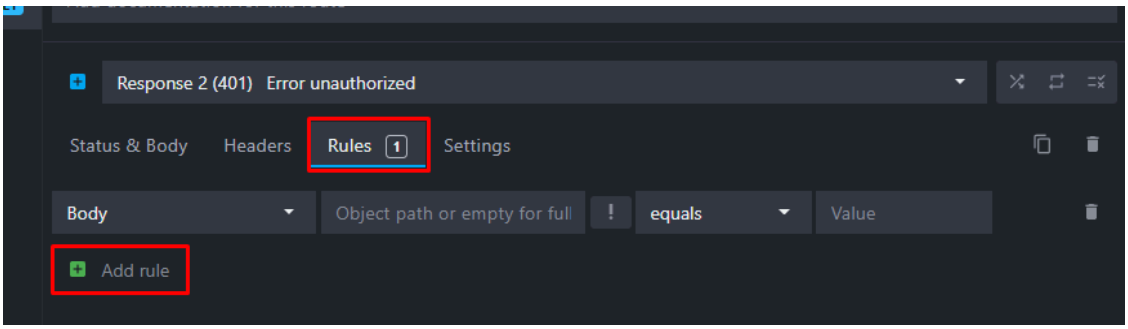
Đổi mã lỗi của response thành 401. Thêm mô tả nếu muốn và cấu hình body trả về ở vùng phía dưới.



Step 3. Thêm rules

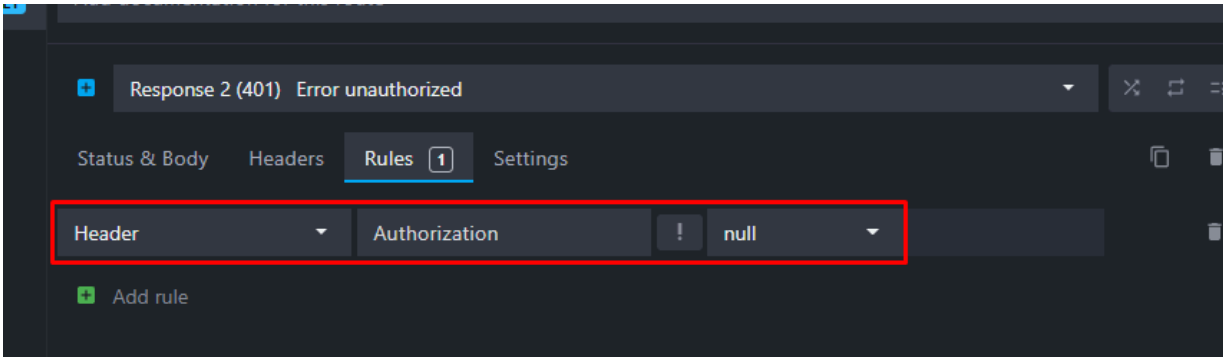
Cuối cùng là thêm rule cho response mới thêm để trả về chỉ khi không có header Authorization.

Nhấn vào nút Add Rule để thêm 1 rule mới.



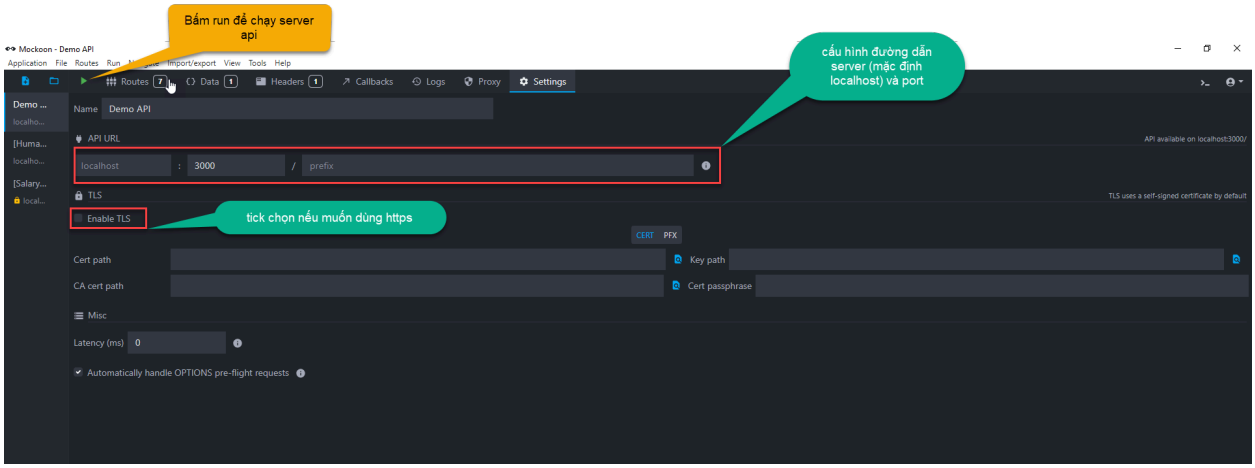
In the new rule, select "Header" and enter "Authorization" as the "Header name".

Ở rule mới, chọn loại Headers, nhập Header Name là Authorization. Sau đó chọn null thay vì equal



4. Chạy server và test trên Postman

a. Cấu hình đường dẫn link server (mặc định là localhost)



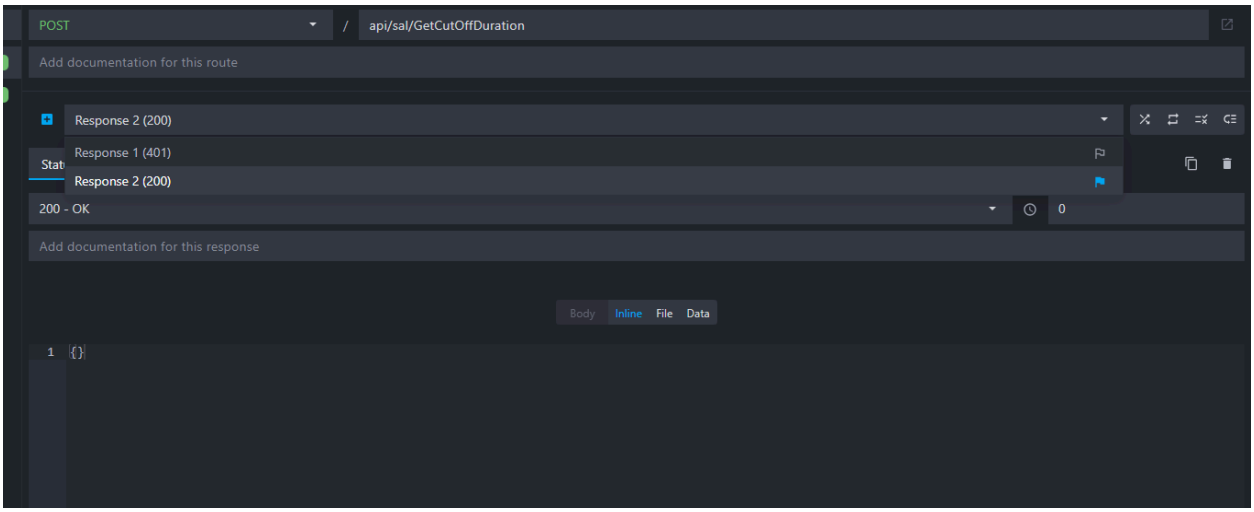
b. Cấu hình port: cấu hình port bất kỳ

c. Tick chọn TLS nếu muốn dùng https

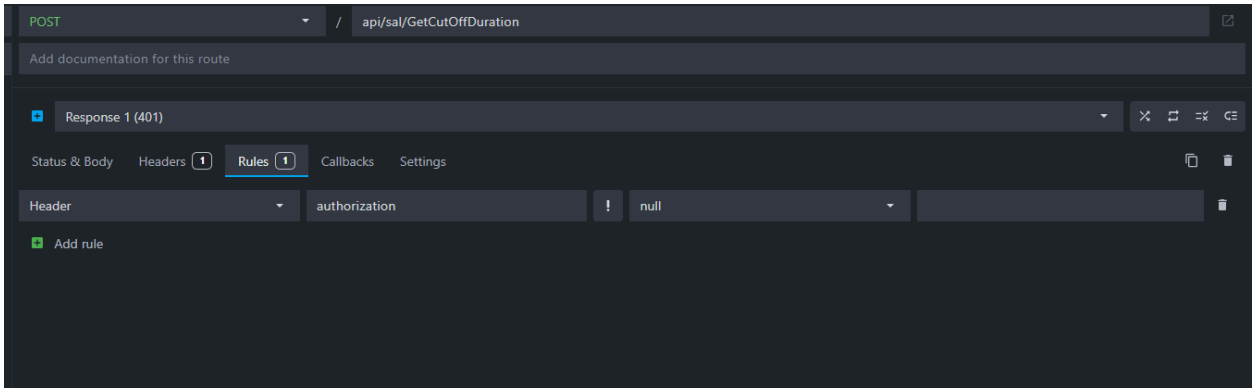
d. Nhấn nút run

e. Test trên postman

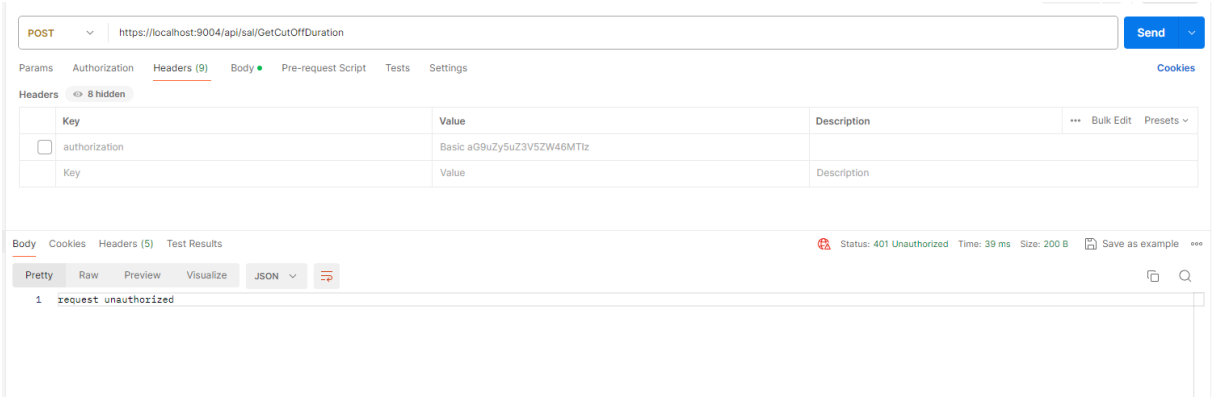
- Ví dụ route cấu hình 2 response, gắn flag ở response 200 để luôn trả về response này khi không cấu hình rule hoặc không có rules nào match.



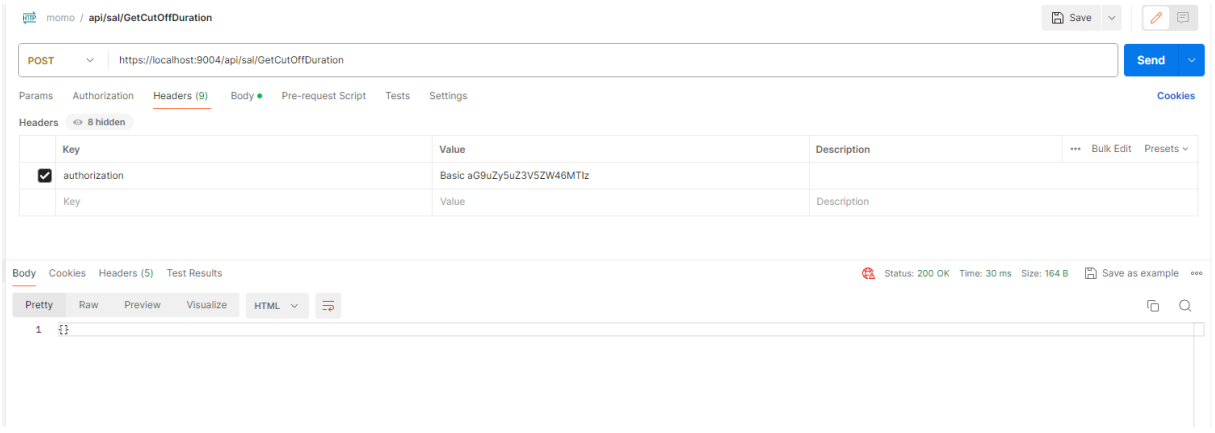
- Response 401 cấu hình rule check header Authorization = null



- Khi request không thêm header Authorization, khớp với rule Authorization = null của response 401.

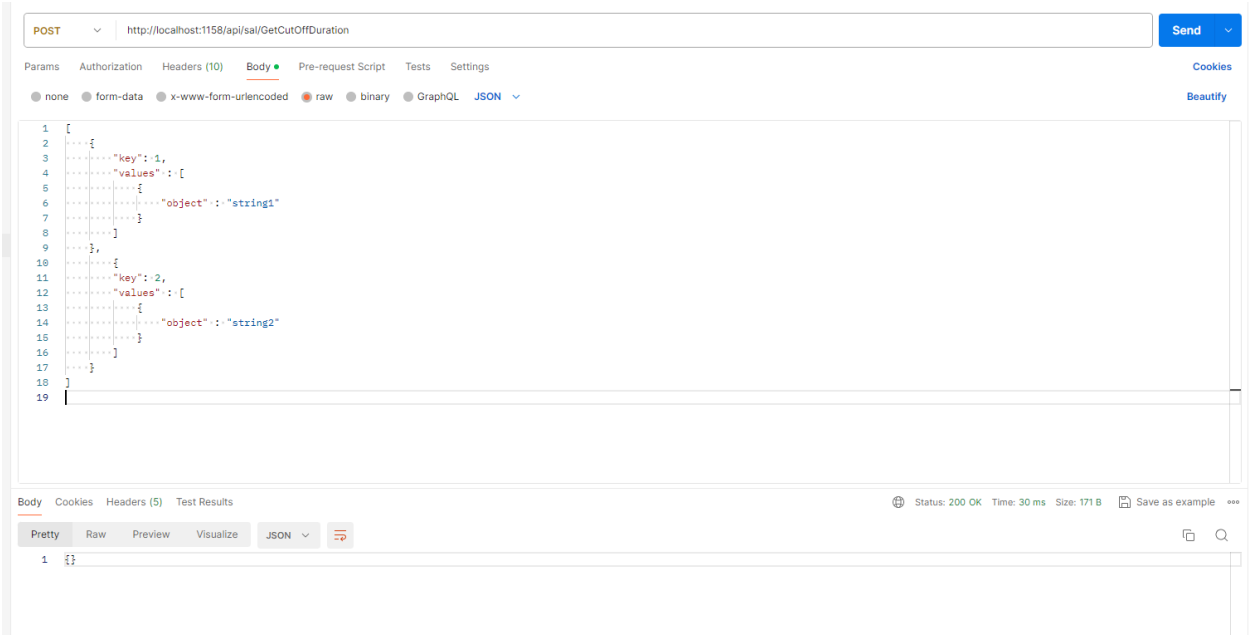


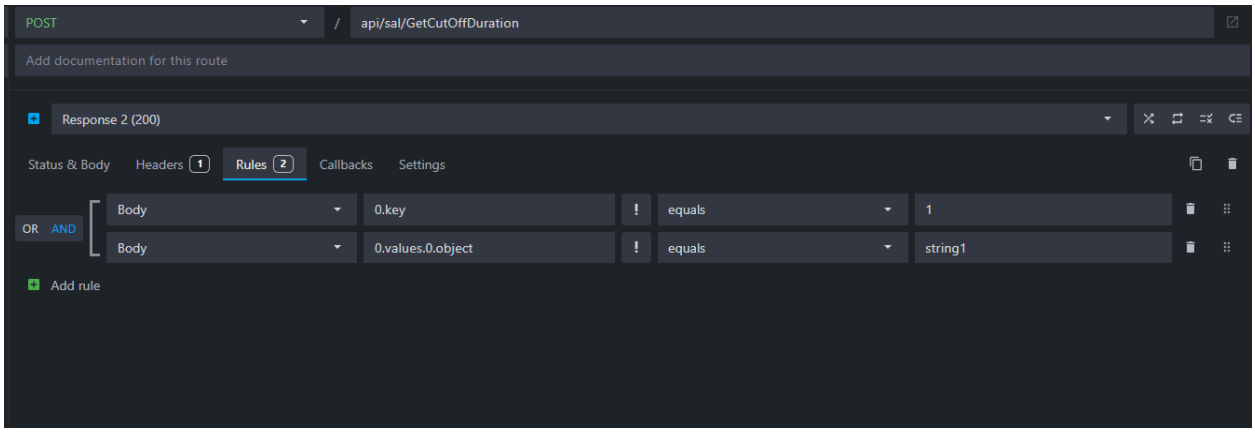
- Khi request có truyền headers Authorization, không rule của response nào match thì sẽ trả về response mặc định là 200.



5. Cấu hình Rules

a. Cấu hình đường dẫn tới property trong request





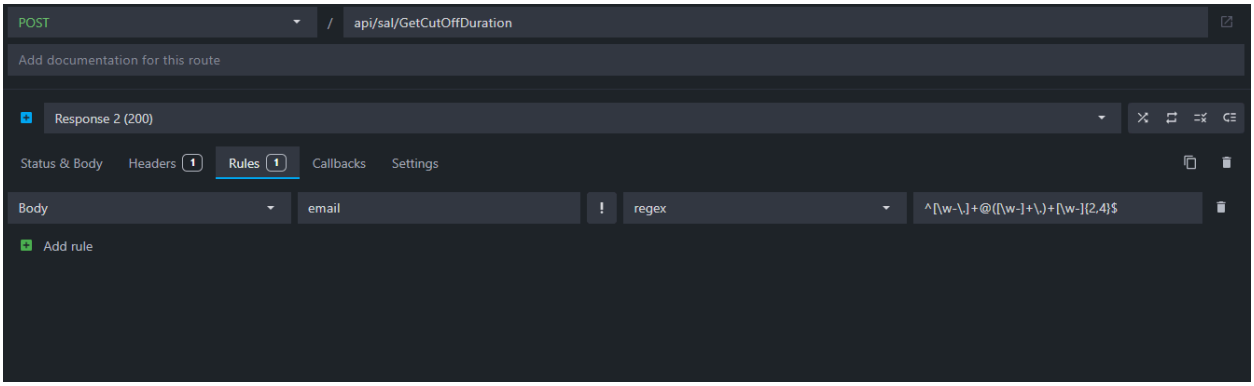
b. Validate độ dài

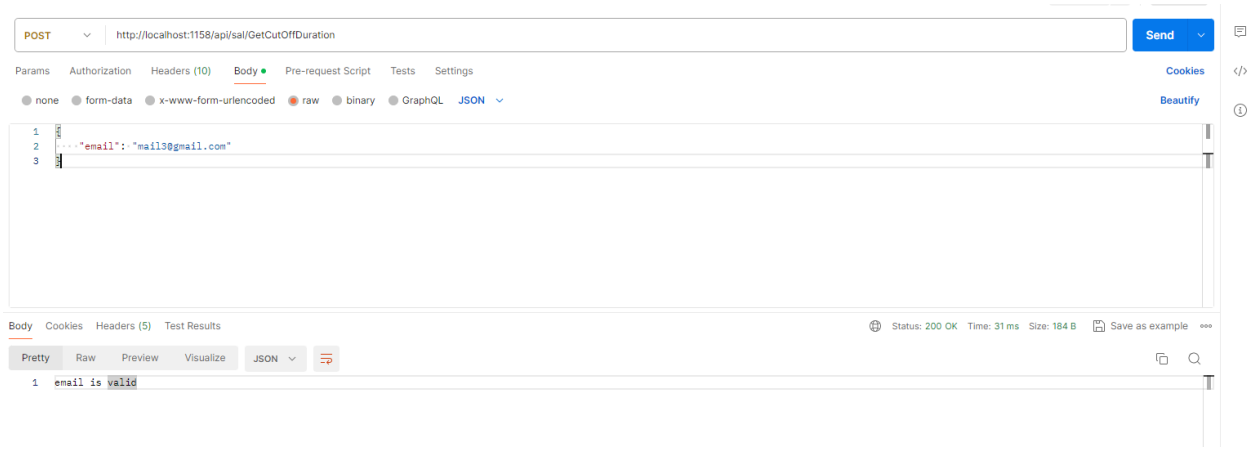
Thêm ký tự vào trong ngoặc vuông nếu string có chứa các ký tự khác ngoài chữ cái và số.

```
^[0-9a-zA-Z]{3,7}$
```

c. Validate Email

```
^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$
```





d. Validate Phone

```
^\+?[1-9][0-9]{7,14}$
```

e. Validate Guid

Check giá trị field là GUID

```
^(?:\{{0,1}(?:[0-9a-fA-F]){8}-(?:[0-9a-fA-F]){4}-(?:[0-9a-fA-F]){4}-(?:[0-9a-fA-F]){4}-(?:[0-9a-fA-F]){12}\}{0,1})$
```

KỊCH BẢN MẪU

Kịch bản mẫu là một tài liệu mô tả chi tiết các bước thực hiện một quy trình hoặc nhiệm vụ cụ thể. Sử dụng để hướng dẫn người dùng cách thực hiện một công việc cụ thể. Bên dưới là một kịch bản mẫu cho “chức năng đăng nhập một ứng dụng cụ thể”

Kịch bản mẫu

Chức năng: đăng nhập

STT	Mô tả	Dữ liệu đầu vào	Dữ liệu đầu ra mong muốn	Kết quả thực tế
TC01	Kiểm tra đăng nhập thành công với username và password hợp lệ	Username hợp lệ, Password hợp lệ	Mã trạng thái 200, Token hợp lệ	
TC02	Kiểm tra hệ thống báo lỗi khi nhập username không hợp lệ	Username không hợp lệ, Password hợp lệ	Mã trạng thái 401, Lỗi "Username không hợp lệ"	
TC03	Kiểm tra hệ thống báo lỗi khi nhập password không hợp lệ	Username hợp lệ, Password không hợp lệ	Mã trạng thái 401, Lỗi "Password không hợp lệ"	
TC04	Kiểm tra hệ thống báo lỗi khi bỏ trống username	Username: "", Password hợp lệ	Mã trạng thái 400, Lỗi "Vui lòng nhập username"	
TC05	Kiểm tra hệ thống báo lỗi khi bỏ trống password	Username hợp lệ, Password: ""	Mã trạng thái 400, Lỗi "Vui lòng nhập password"	
TC06	Kiểm tra hệ thống báo lỗi khi nhập username và password sai định dạng	Username: "abc123", Password: "12345"	Mã trạng thái 400, Lỗi "Username/password không hợp lệ"	
TC07	Kiểm tra hệ thống chống tấn công SQL injection	Username: ' OR 1=1; --, Password: "abc123"	Mã trạng thái 400, Lỗi "Lỗi SQL injection"	
TC08	Kiểm tra hệ thống chống tấn công XSS	Username: "<script>alert('XSS attack');</script>", Password: "abc123"	Mã trạng thái 400, Lỗi "Lỗi XSS"	

Kịch bản 1: Chức năng đăng nhập:

Mục tiêu:

- Tạo API giả lập cho chức năng đăng nhập
- Kiểm thử chức năng đăng nhập của ứng dụng

Đối tượng:

- Nhân viên triển khai
- Nhân viên kiểm thử

Công cụ:

- Mockoon
- Postman

Bước 1: Tạo API giả lập trong Mockoon

1. Truy cập trang web Mockoon: https://mockoon.com/
2. Tạo dự án mới
3. Thêm API mới
4. Chọn phương thức HTTP: POST
5. Chọn URL: /api/login
6. Request , Response, cấu hình phản hồi

Request - Yêu cầu			
STT	**Tên trường**	**Kiểu dữ liệu**	**Ghi Chú**
1	username	string	Tên đăng nhập
2	password	string	Mật khẩu
Response - Phản hồi			
1	succcess	boolean	
2	message	string	
Response Config - Cấu hình phản hồi			

1	Valid	Nếu username và password hợp lệ, trả về success = true và message = "Đăng nhập thành công"
2	Invalid	Nếu username hoặc password không hợp lệ, trả về success = false và message = "Đăng nhập thất bại"

Bước 2: Gửi yêu cầu và kiểm tra phản hồi trong Postman

1. Mở Postman
2. Chọn phương thức HTTP: POST
3. Nhập URL: http://localhost:3000/api/login (thay đổi localhost nếu cần thiết)
4. Thêm body JSON với dữ liệu yêu cầu:
5. JSON

```
{
  "username": "your_username",
  "password": "your_password"
}
```

6. Gửi yêu cầu
7. Kiểm tra phản hồi:

Kịch bản đăng nhập vào hệ thống				
STT	**Mô tả**	**Dữ liệu đầu vào**	**Đầu ra**	**Kết quả thực tế**
TC01	Kiểm tra đăng nhập thành công với username và password hợp lệ	Username hợp lệ, Password hợp lệ		phản hồi sẽ có success = true và message = "Đăng nhập thành công"
TC02	Kiểm tra hệ thống báo lỗi khi nhập username không hợp lệ	Username không hợp lệ, Password hợp lệ		phản hồi sẽ có success = false và message = "Đăng nhập thất bại"

Kịch bản 2: Chức năng AAA: