Swagger để làm gì ?

Document api đọc api từ controller để đưa web tĩnh domain/swagger

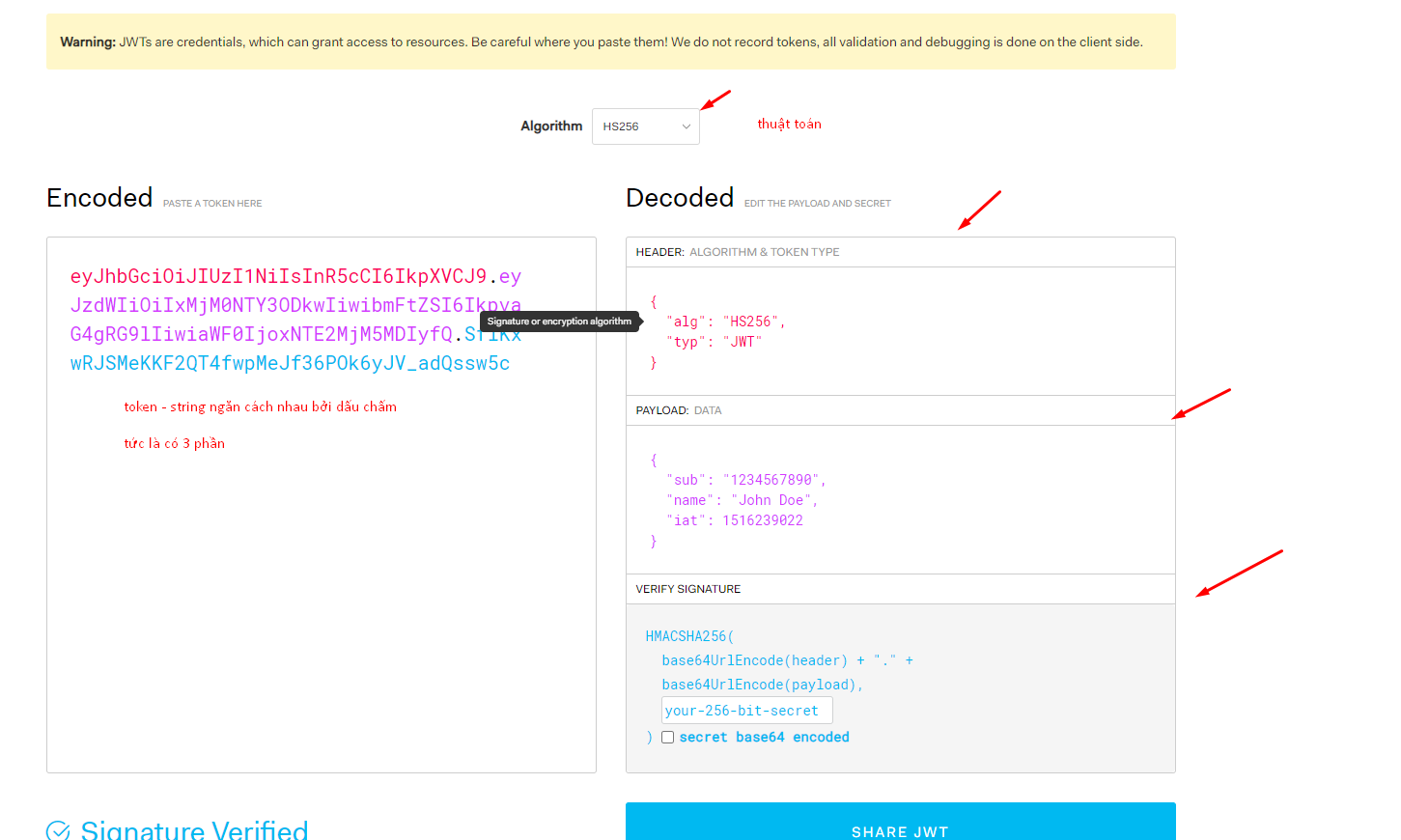
~ postman

Tạo api login

Đầu vào là username + password

Đầu ra : token (JWT) => chuỗi gồm 3 thành phần ngăn cách nhau bởi dấu .

Trang web: <https://jwt.io/>



Tóm lại token: chuỗi chứa phần header, payload và veryfy signature

Chúng ta chỉ quan tâm đến 2 thành phần là header, payload

Token => header: chuyển đổi thành 1 object dạng json

Payload: cũng dạng là json

Thường có các thông tin mà chúng ta gắn vào, thời gian hết của token

Json là kiểu dữ liệu gồm Key và Value

Tức là có dạng: {

“key1” : “value1”,

“key2” : “value2”,

…

}

Các web hiện nay thì gần như đều tương tác dữ liệu dưới dạng là json

Làm thế nào để chúng ta biết được web của mình đã đăng nhập

Bằng đơn giản là chúng ta nhập username + password và bắn lên server dưới dạng là json

Với phương thức là POST

Chúng ta nhận token về và giải mã cái token đó để xác nhận là chúng ta đã đăng nhập thành công.

***API là gì?***

* **Application Programming Interface** – giao diện lập trình ứng dụng

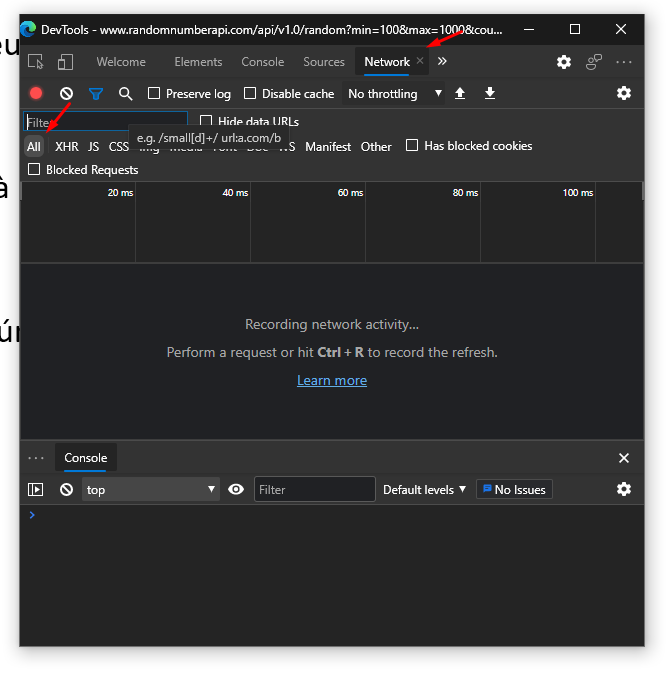
Dùng để làm gì?

* Giao tiếp giữa các nền tảng khác nhau
* Giao tiếp giữa các ứng dụng khác nhau
* Giao tiếp giữa các con web với nhau, giữa web và mobile, giữa mobile và desktop… thông qua API
* Tương tác qua kiểu dữ liệu là json

1 API sẽ có những gì ?

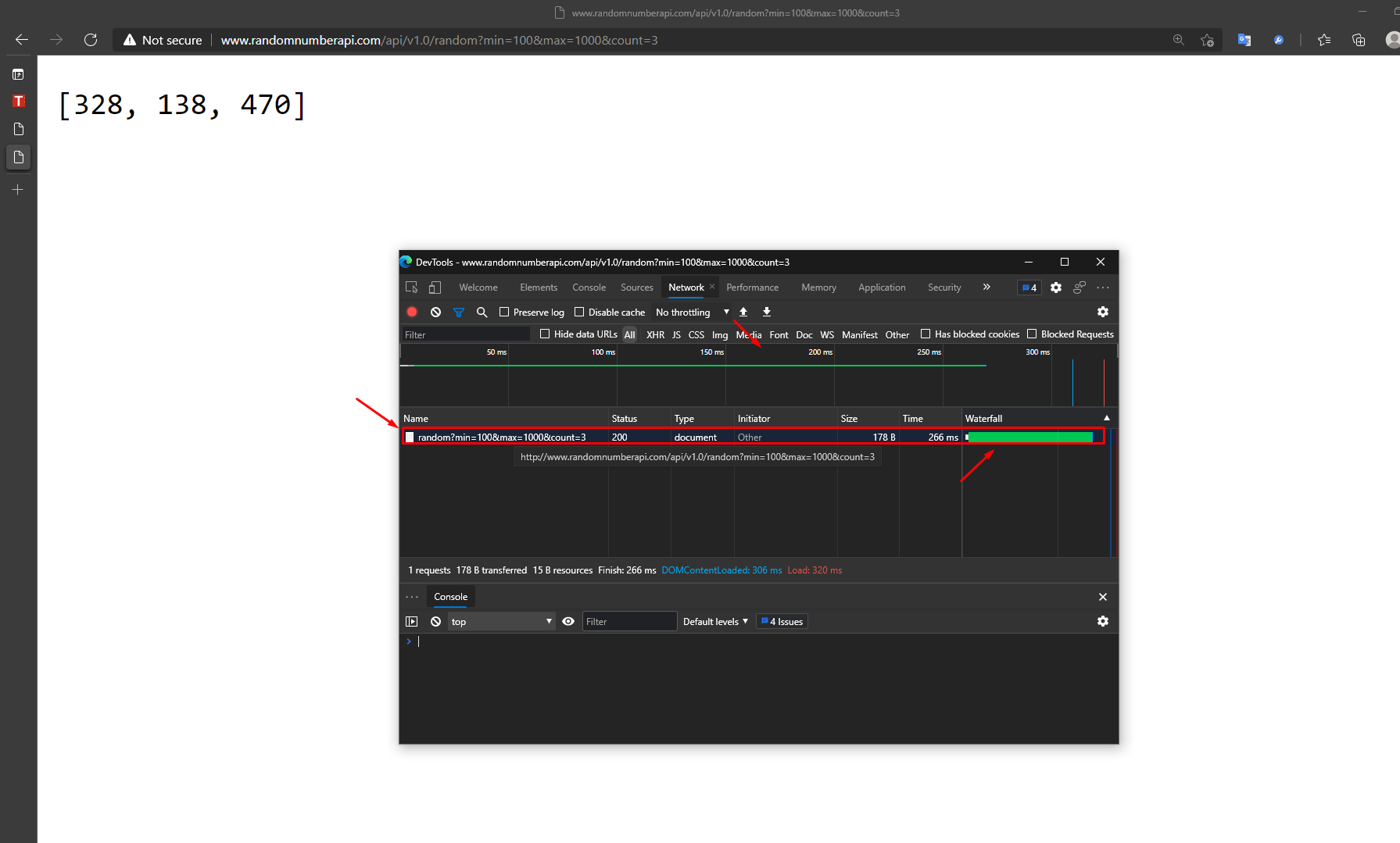
Và làm sao chúng ta nhận được đấy là 1 Api ???

Ví dụ:

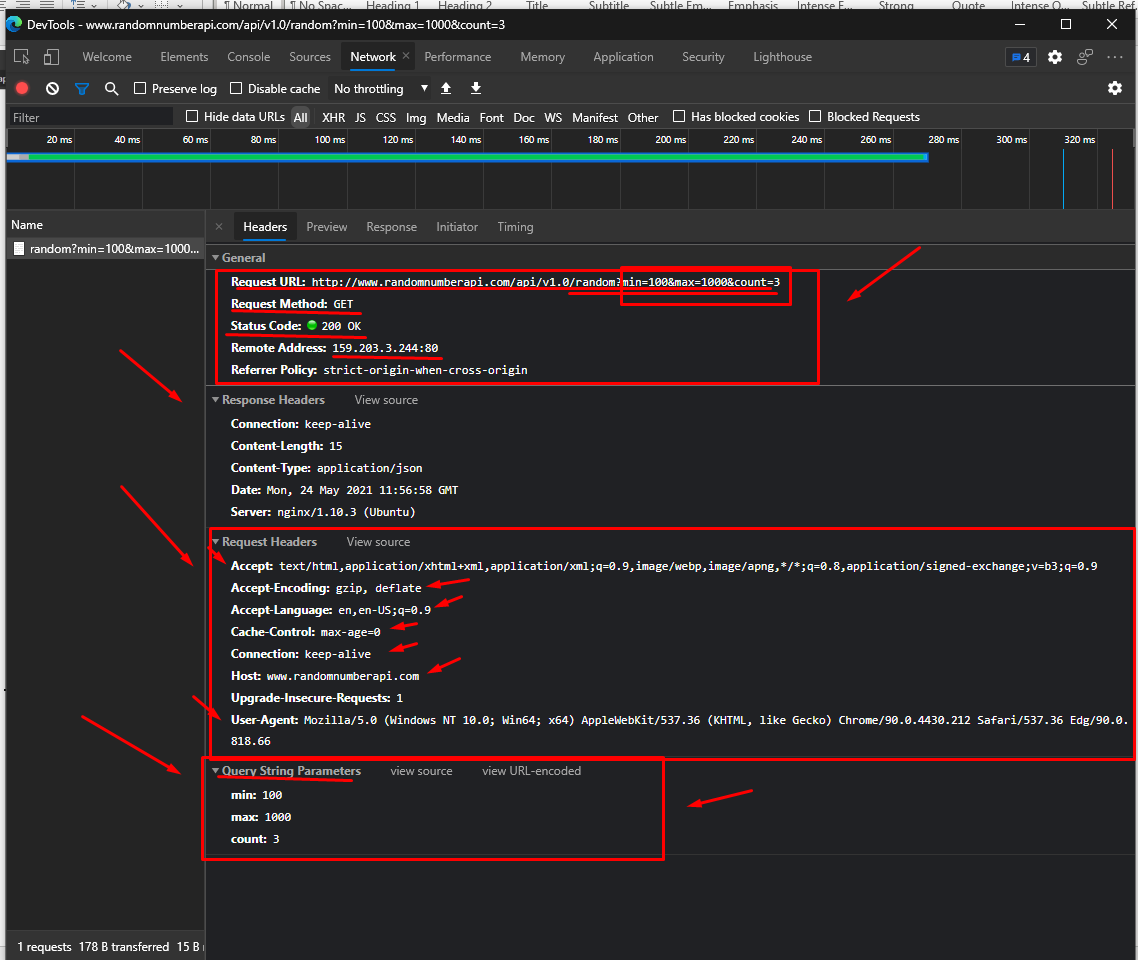


Cách kiểm tra API của 1 trang web chúng ta bật F12 để vào tab network

Và kiểm tra xem tên api là gì? , url của api là gì?, dữ liệu được bắn lên



Thanh màu xanh là thanh chỉ ra rằng thời gian từ lúc gửi request đến lúc api đó nhận phản hồi từ server



Chúng ta có tổng 4 phương thức: GET, POST, PUT, DELETE,

Gồm cả request URL,

Status code: 200

Status code đọc trong đây: <https://topdev.vn/blog/http-status-code-la-gi/>

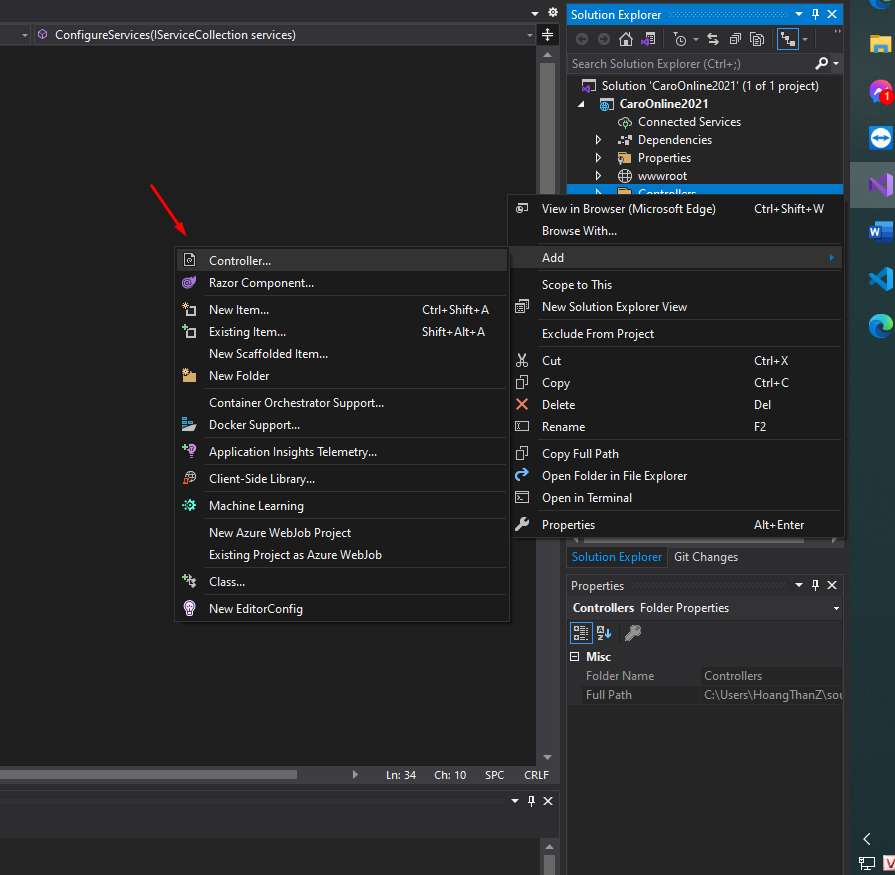
Cách viết tạo ra một API login như thế nào?

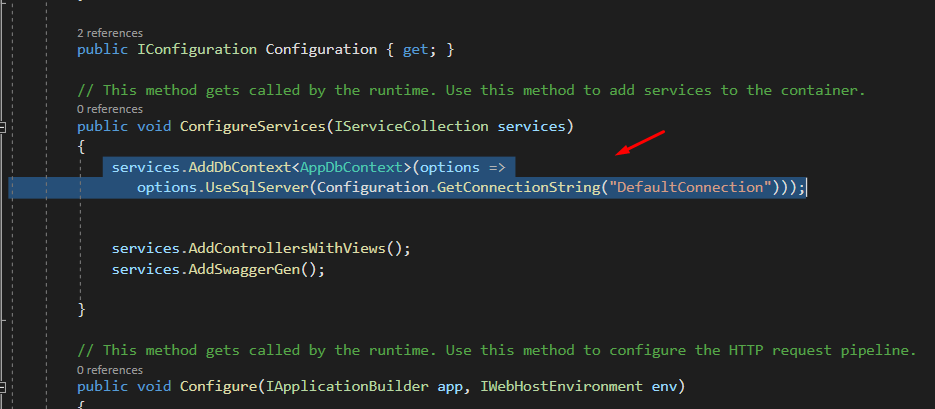
Yêu cầu là phải biết SQL, Linq, Entity framework, C#

Bước 1: tạo controller và định nghĩa phương thức của api, đối số truyền vào mà api cần là gì, kiểu dữ liệu mà api sẽ trả về

Bước 2: Thực hiện truy vấn với các câu lệnh của linq

Bước 3: test thử nghiệm trên swagger or postman (nếu có)



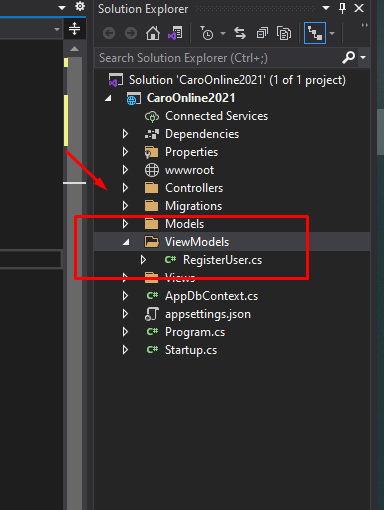


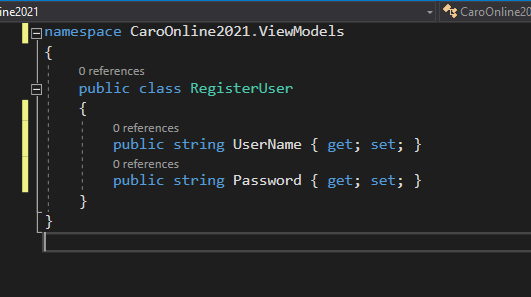
Chúng ta biết rằng trong file startup.cs có services.AdđBContext<AppDbContext>….

Nó được rằng là đây là class về thao tác dữ liệu mà dùng chung cho toàn hệ thống.

Và chỉ việc inject nó vào trong contructor của mỗi controller mà mình cần dùng.

Tạo ra thư mục ViewModels để chứa các class mà dùng để gửi hoặc nhận dữ liệu





Đồng nghĩa với việc chuyển sang json sẽ ở dạng

{

“userName”: “ <tùy chúng ta điền>”,

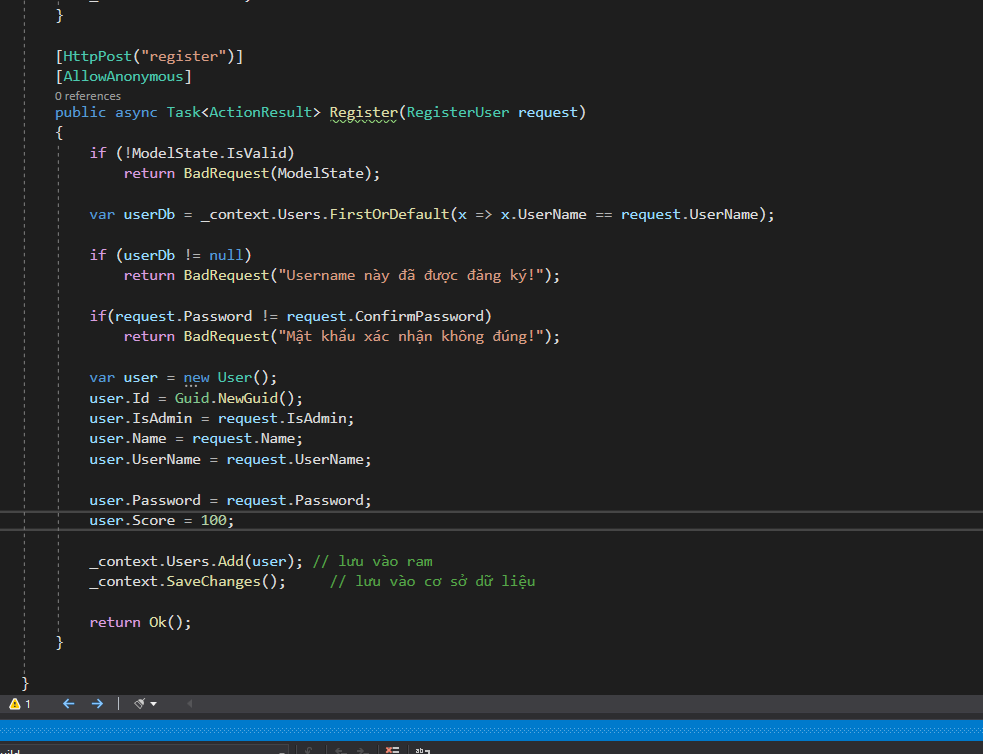
“passwork” : “<tùy chúng ta điền>”

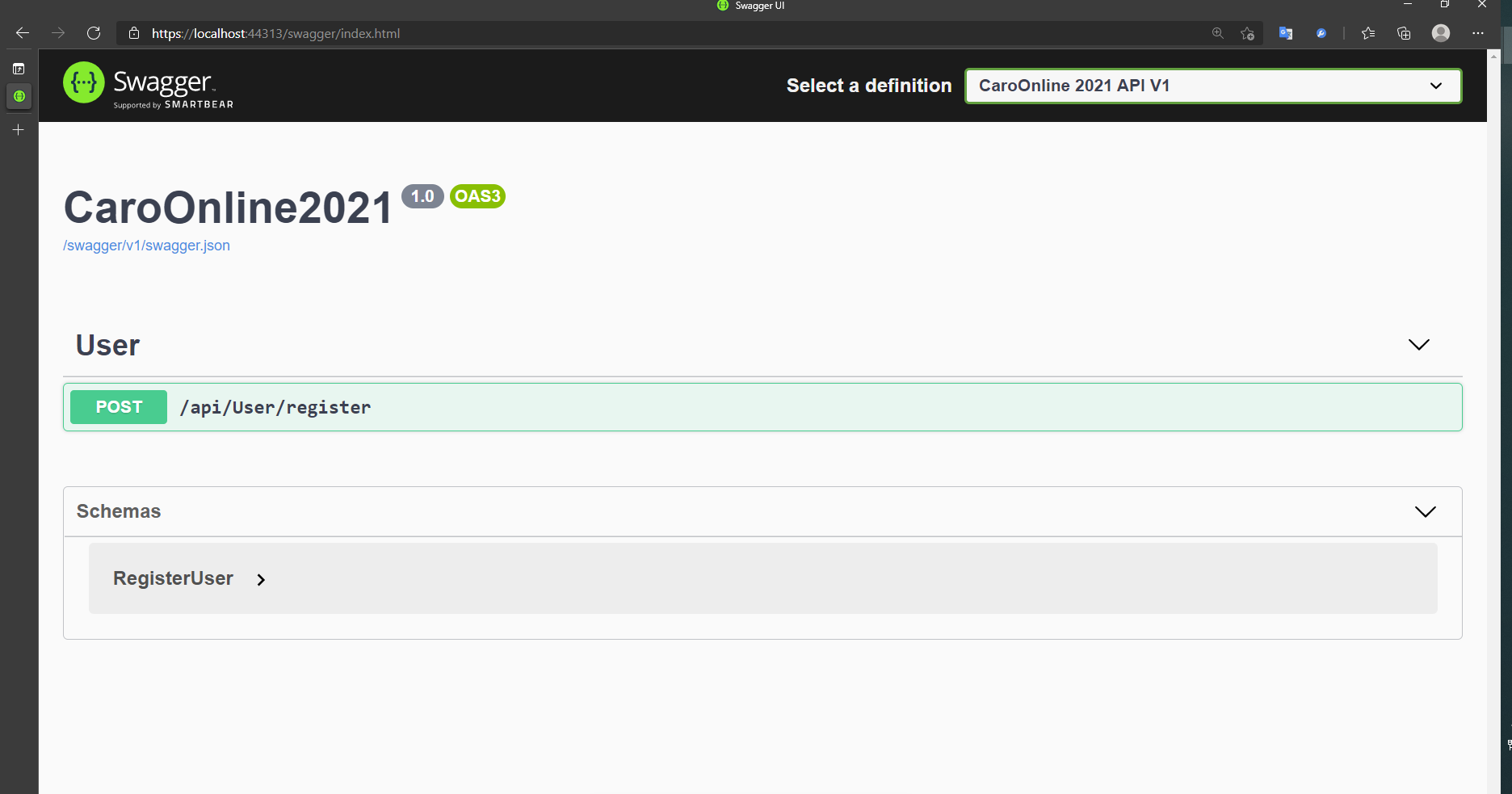
}

**GUID** (hoặc **UUID**) **là** từ viết tắt của 'Định danh duy nhất toàn cầu' (hoặc 'Định danh duy nhất toàn cầu'). Nó **là** số nguyên 128 bit được sử dụng để xác định tài nguyên. Thuật ngữ **GUID** thường được sử dụng bởi các nhà phát triển làm việc với các công nghệ của Microsoft, trong khi **UUID** được sử dụng ở mọi nơi khác.2 thg 9, 2008

Ví dụ: 55cea40c-b3e6-425f-83ff-8e834147169c (trông nó như này)

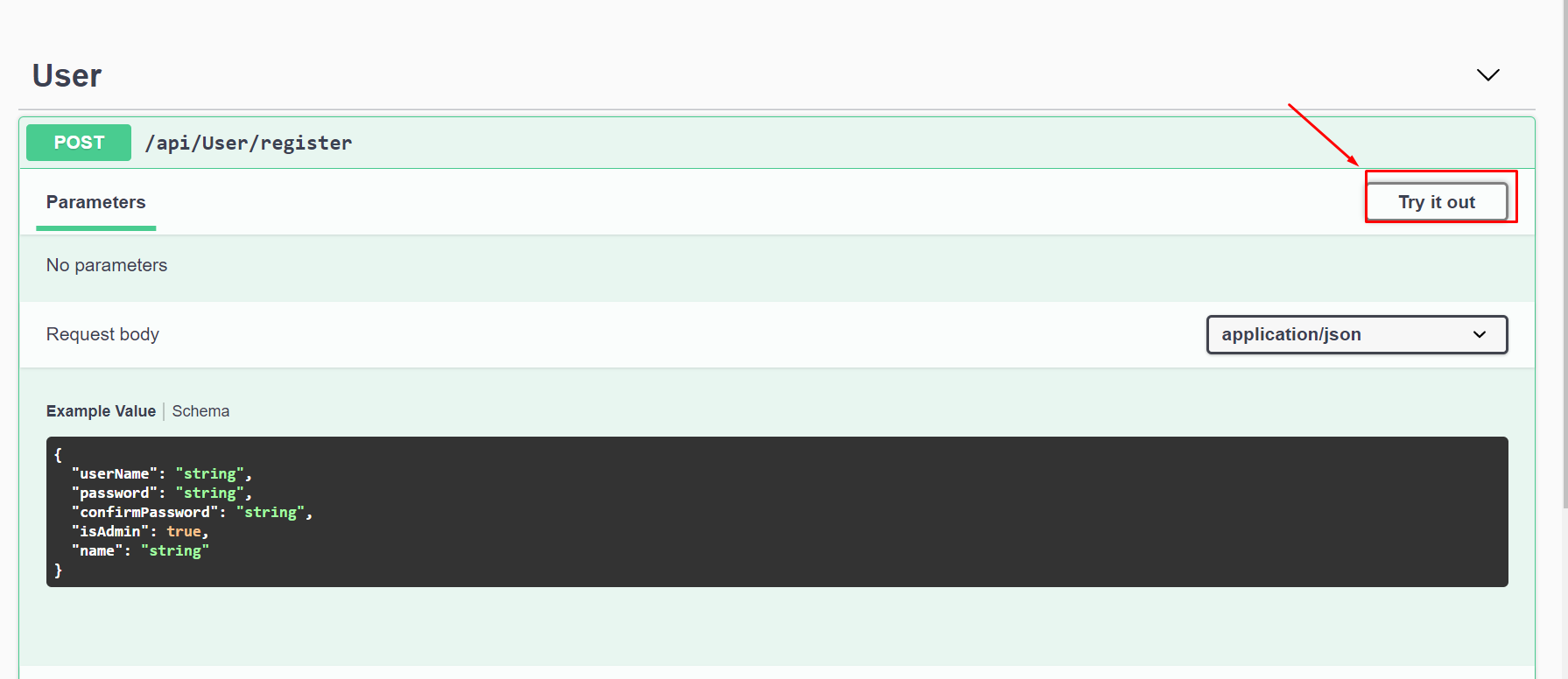
358c3400-437f-492b-ae42-125e5547fa47 ( để ý)

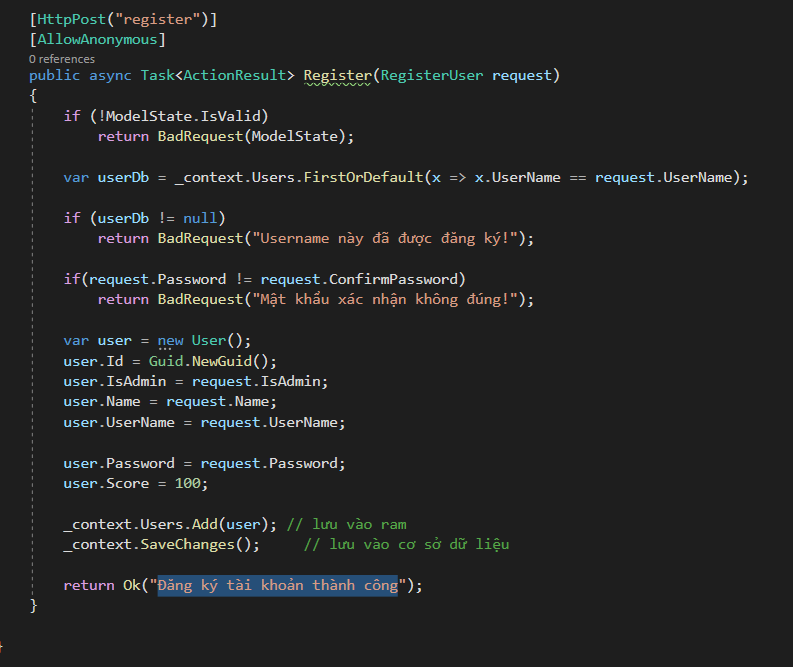




Lưu ý <https://localhost:44313/> cổng 44313 là tùy vào từng máy mà nó tự động sinh ra các cổng là khác nhau

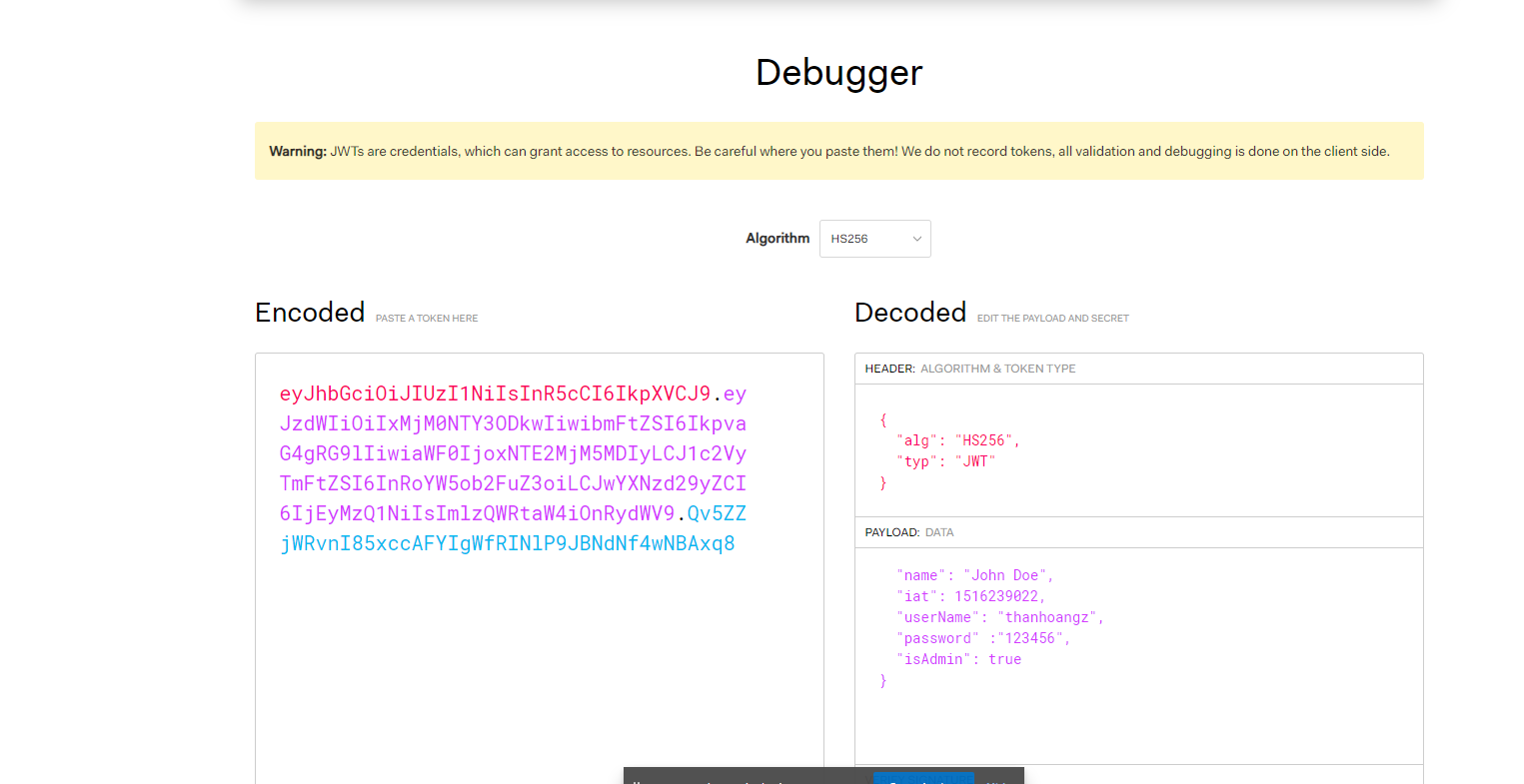
Gõ thêm /swagger và enter để truy cập vào document api (swagger)



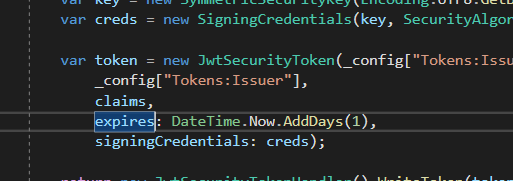


Hoàn thành xong api đăng ký

Bây giờ muốn tạo ra api là login

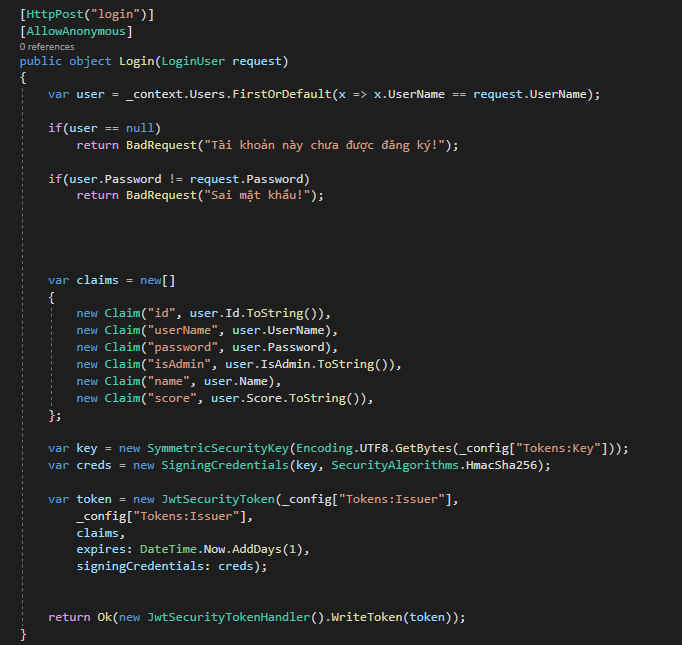


Lưu ý : token có thời gian hết hạn tùy vào chúng ta cấu hình



ở đây minhd đang cấu hình thời gian hết của token này là 1 ngày.

Cách tạo token từ login api



Thực hiện check điều kiện xem xem tài khoản này có tồn tại, check password có đúng

Nếu như không thì trả status code là 400 và 1 chuỗi báo lỗi gì

Claims: là để gắn những thông tin của user vào payload của token(jwt)

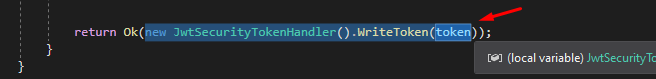
Lấy khóa (là từ 15 ký tự trong file appsetting.json ra để làm khóa bí mật để mã hóa SHA 256

Sau đó chèn token này với các thông tin



Đây là phân token chèn thêm khi kết hợp với khóa và issues trong appseting.json

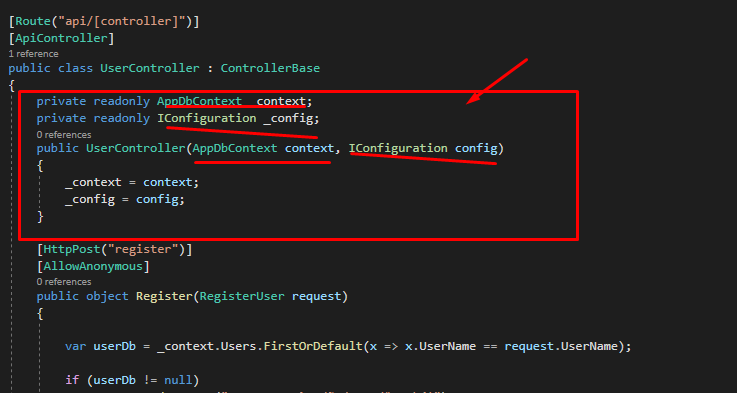
Để đưa ra exp: thời hạn hết token này

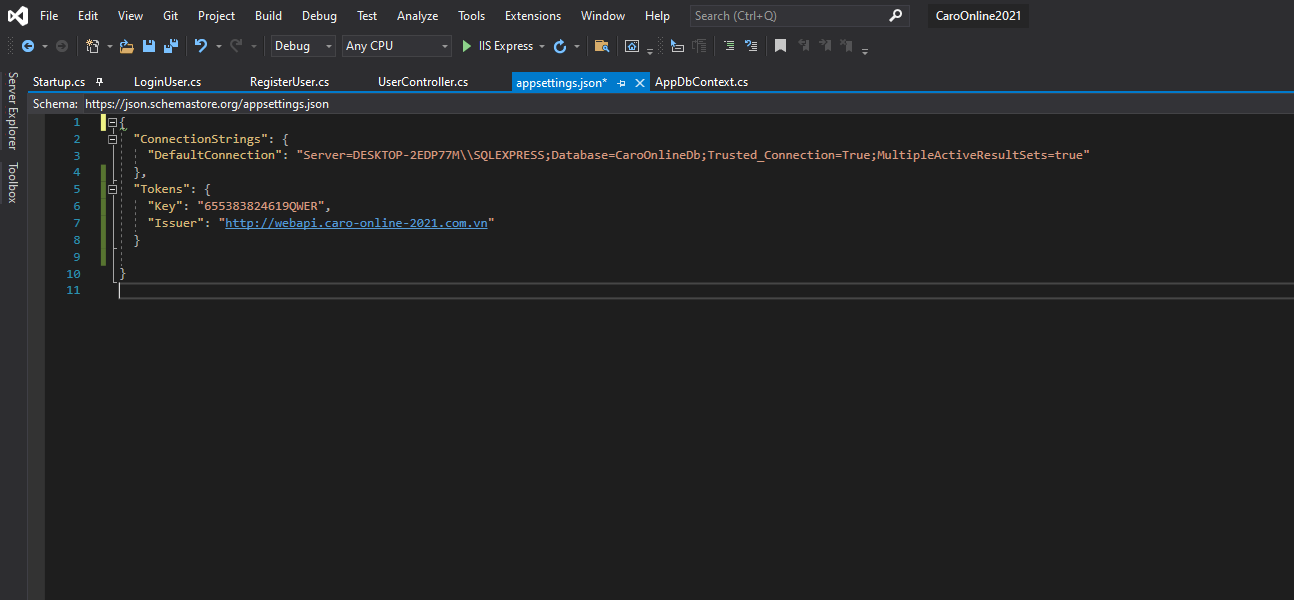


Với đoạn này biến đổi 3 thông tin bên trên + với những thông tin đã chèn vào = claims trước đó để tạo tự động 1 cái token gồm 3 thành phần (xem jwt là gì)

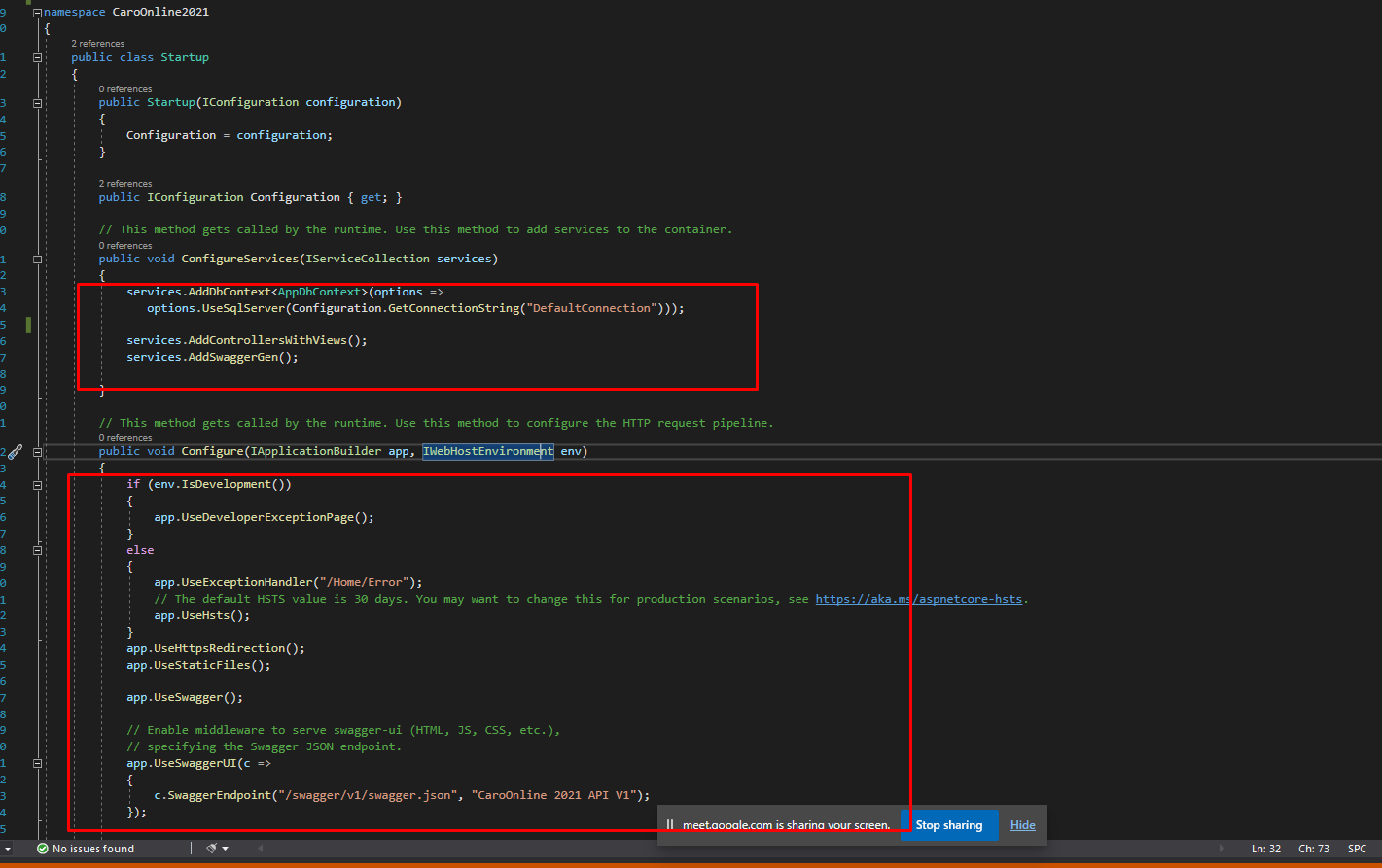
Chúng ta có thể copy cái token mà hệ thống trả ra cho chúng ta để thực hiện encoded lấy được lại các thông tin mà chúng ta đã chèn vào phần payload(data)

Lưu ý một số cấu hình sau:





Còn trong file starup.cs



Như vậy chúng ta đã viết xong 2 api là đăng ký và login cho tài khoản bất kì.