Fresher token sẽ tồn tại trong khoảng thời gian dài

Dùng để lấy lại acess token khi access token hết hạn mà không cần phải đăng nhập lại

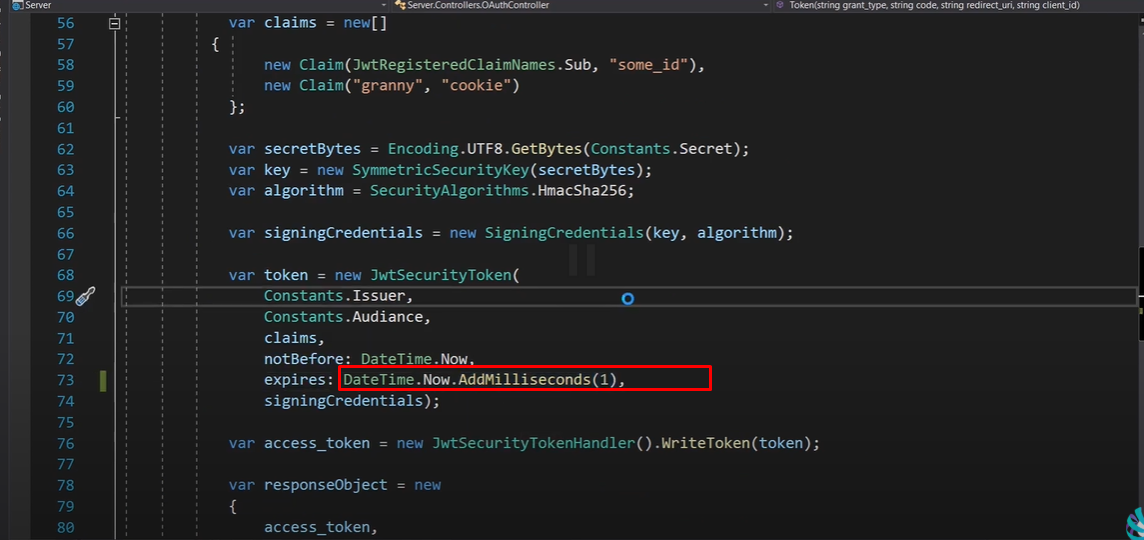
Access token thường dùng để lộ ra front end như cookie hoặc lưu session

Tuy nhiên refresh token phải được bảo mật

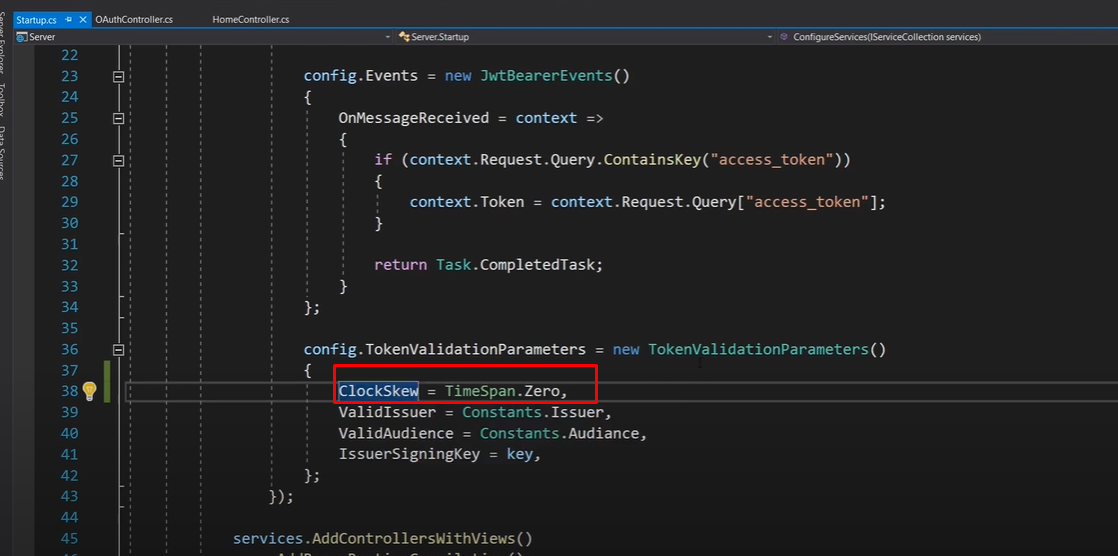
Lý do là refresh token tồn tại trong khoảng thời gian dài

Refreshe token phải được gắn với client isssued nó, mục đích là nếu refresher token bị đánh cắp, thì khi dùng client khác thì không sử dụng được token đó

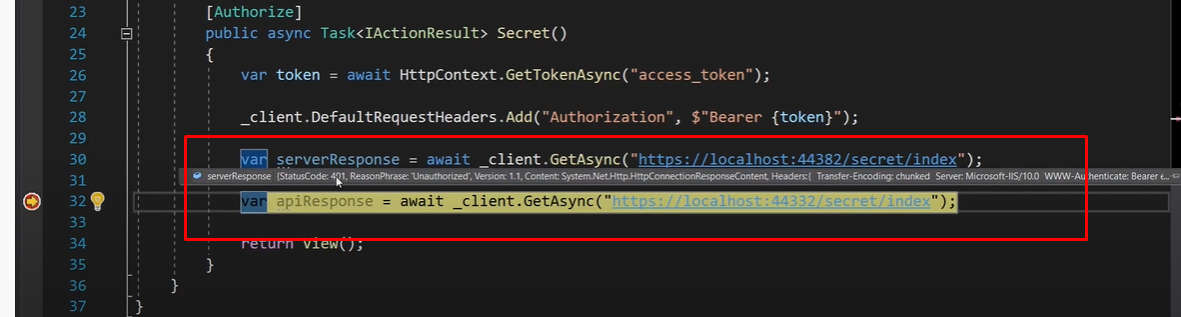
Kịch bản: access token hết hạn, vào link authen bị lỗi 401, dùng refresh token lấy lại token, dùng token đó request vào link authen trả về 200



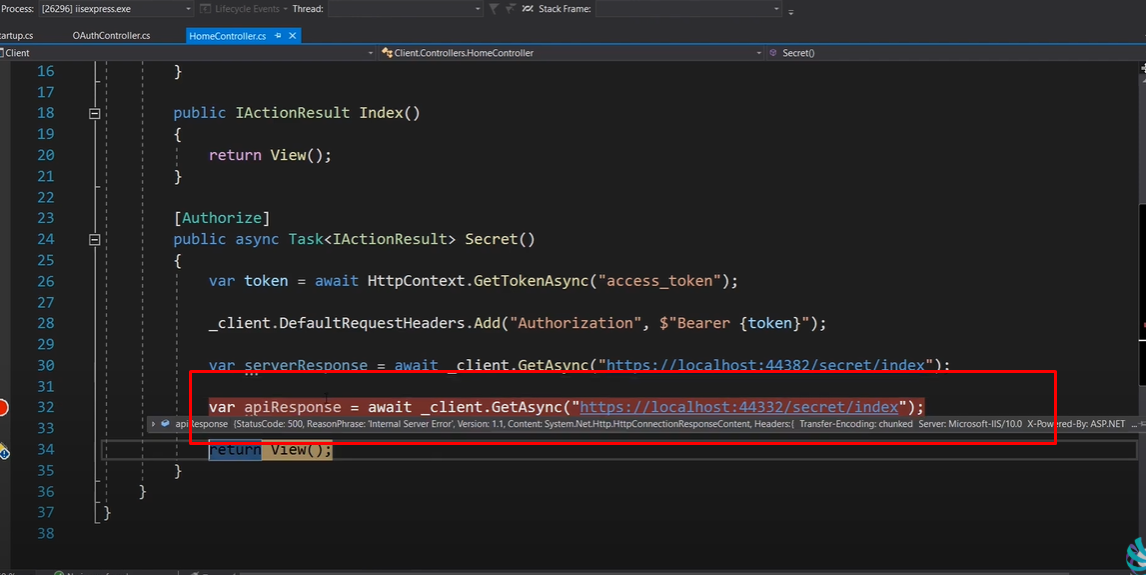
Thay expires thành 1 milisecond, giá trị này nhỏ hơn 5 phút thì nó sẽ là 5 phút



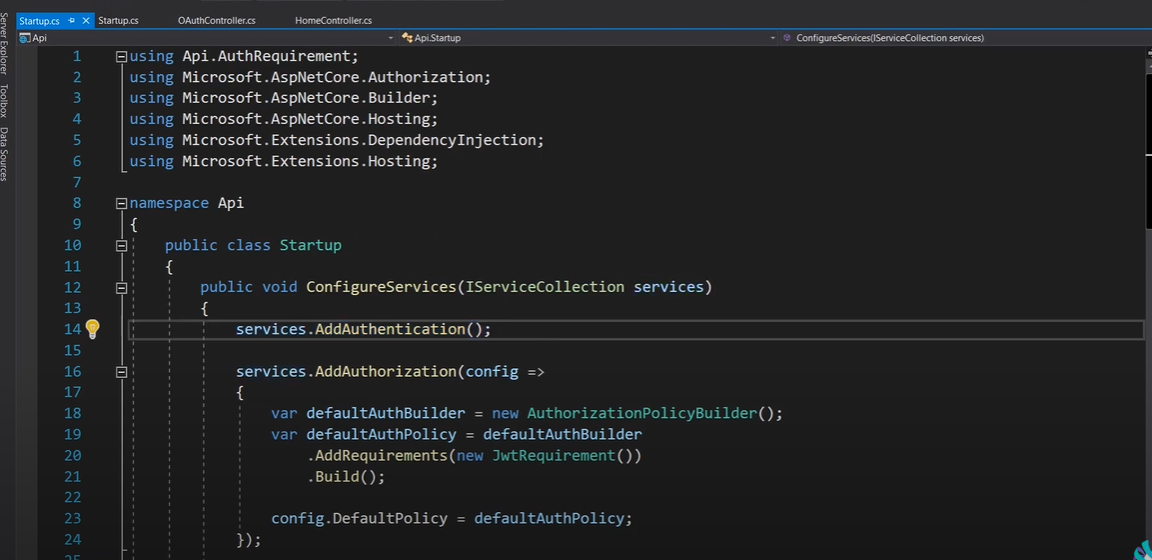
Đưa giá trị skew về zero thì sẽ tính từ 0, Đây là giá trị ở chổ apply token vào client trên startup



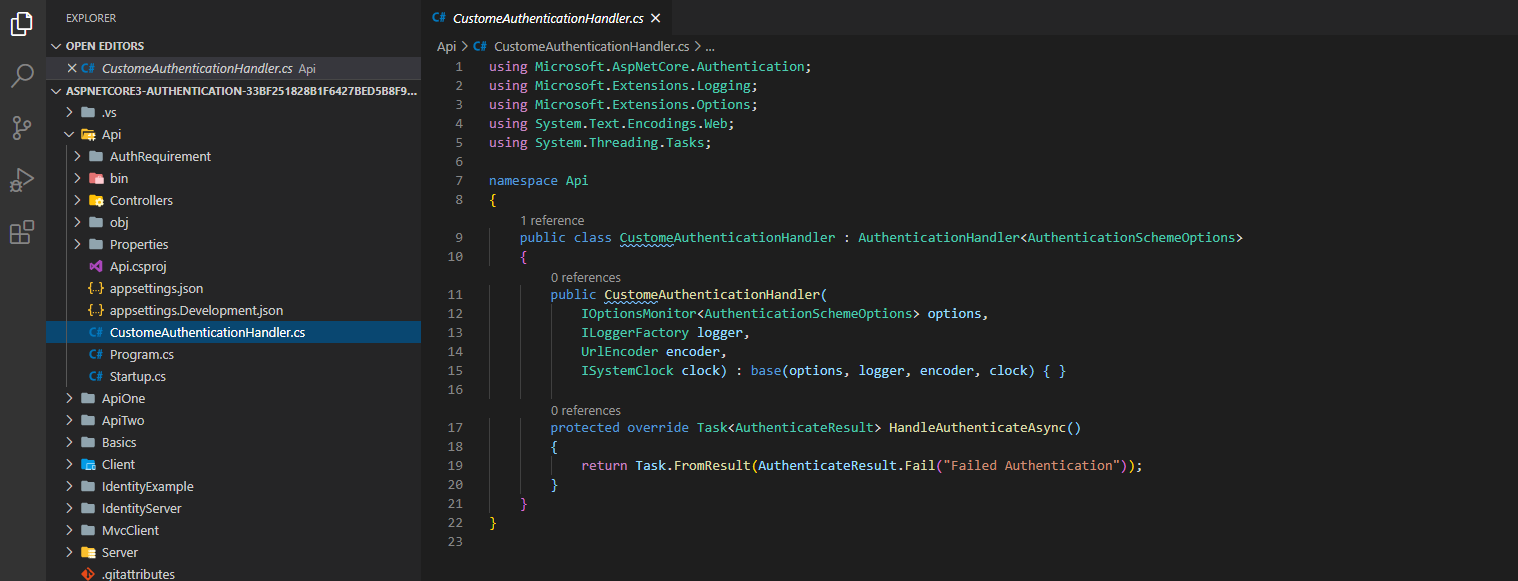
Nhận về lỗi 401 vì token đã hết hạn



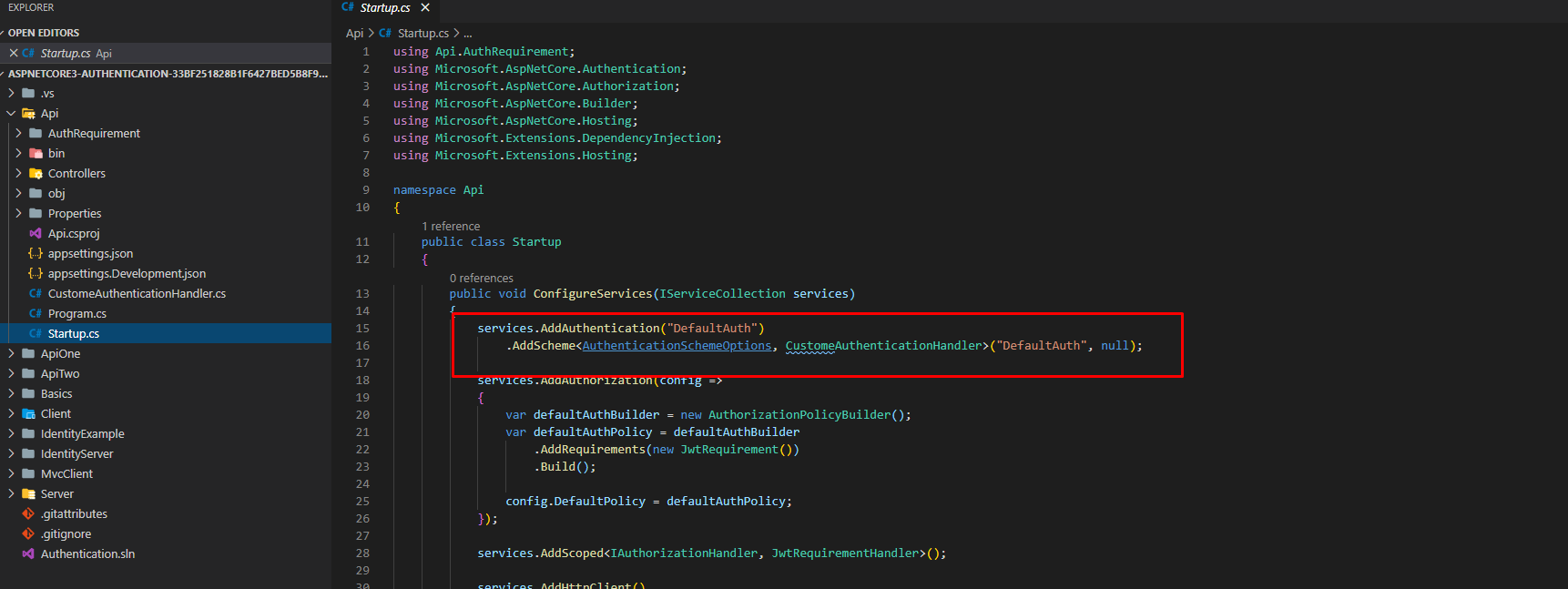
Đáng ra apiResponse phải trả về 401 nhưng ở đây lại 500 -> fix lỗi này



Vào app api tạo custom authetication scheme rồi add vào, hoặc là sửa lại authentication hadler

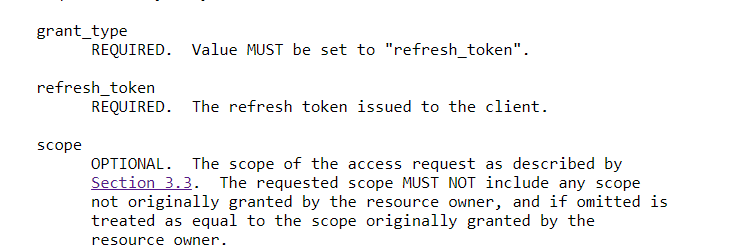


Handler xảy ra khi authentication fail, trả về lỗi fail như trên thì request sẽ trả về 401 thay vì 500

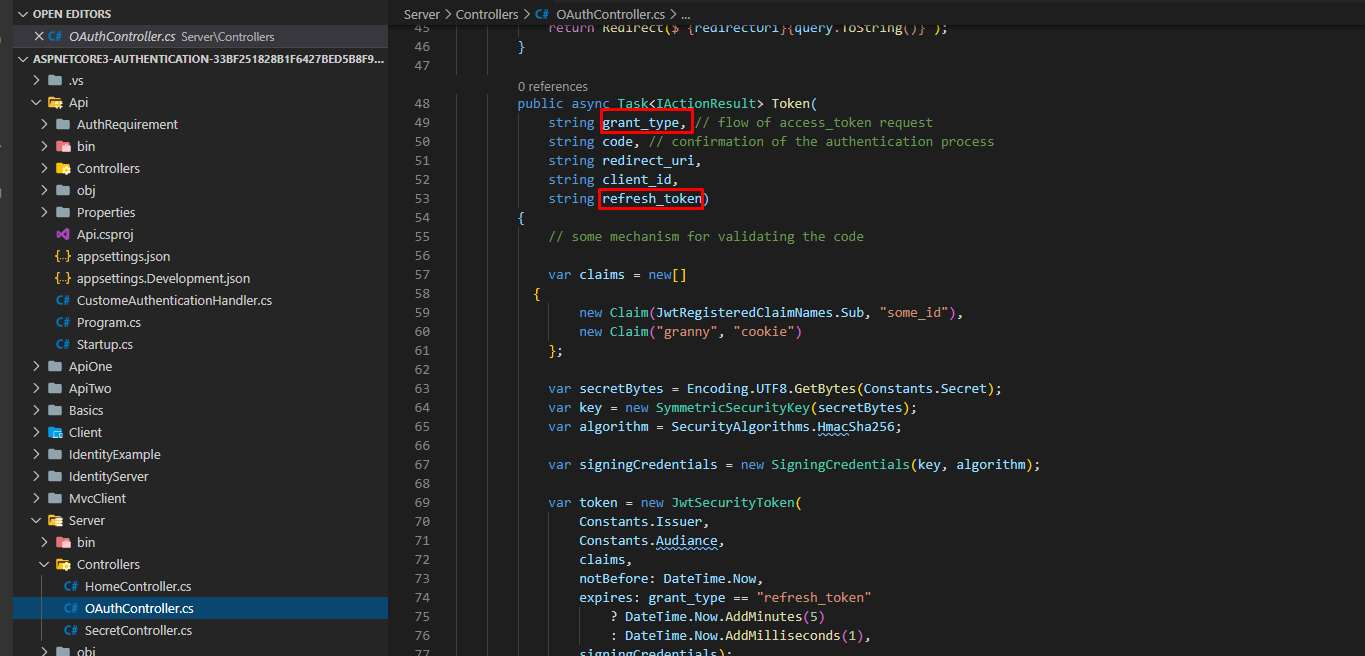


Add vào, kết quả là lỗi authen trên app api sẽ trả về 401 thay vì 500

Sử dụng lại các implemt từ bài trước ( có thể tham khảo source)

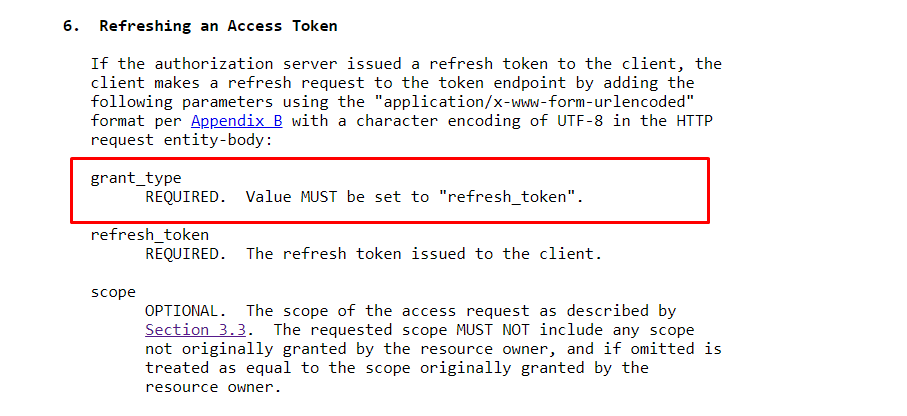


Đây là các biến sử dụng với refresh token, nên thêm các biến này vào, scope là có thể có hoặc không

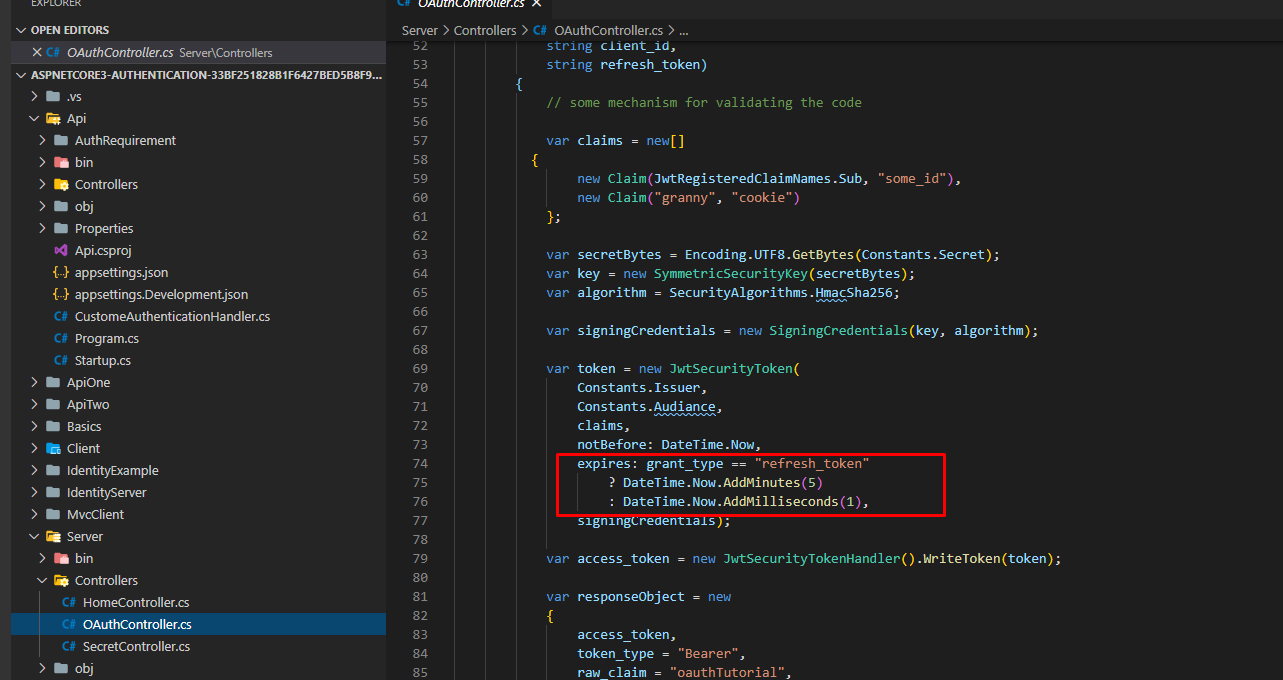


Thêm vào endpoint token (end poitn này sinh ra token khi sử dụng với client xem bài grant)

Khi làm bình thường thì cứ nghĩ đây là hàm sinh ra token

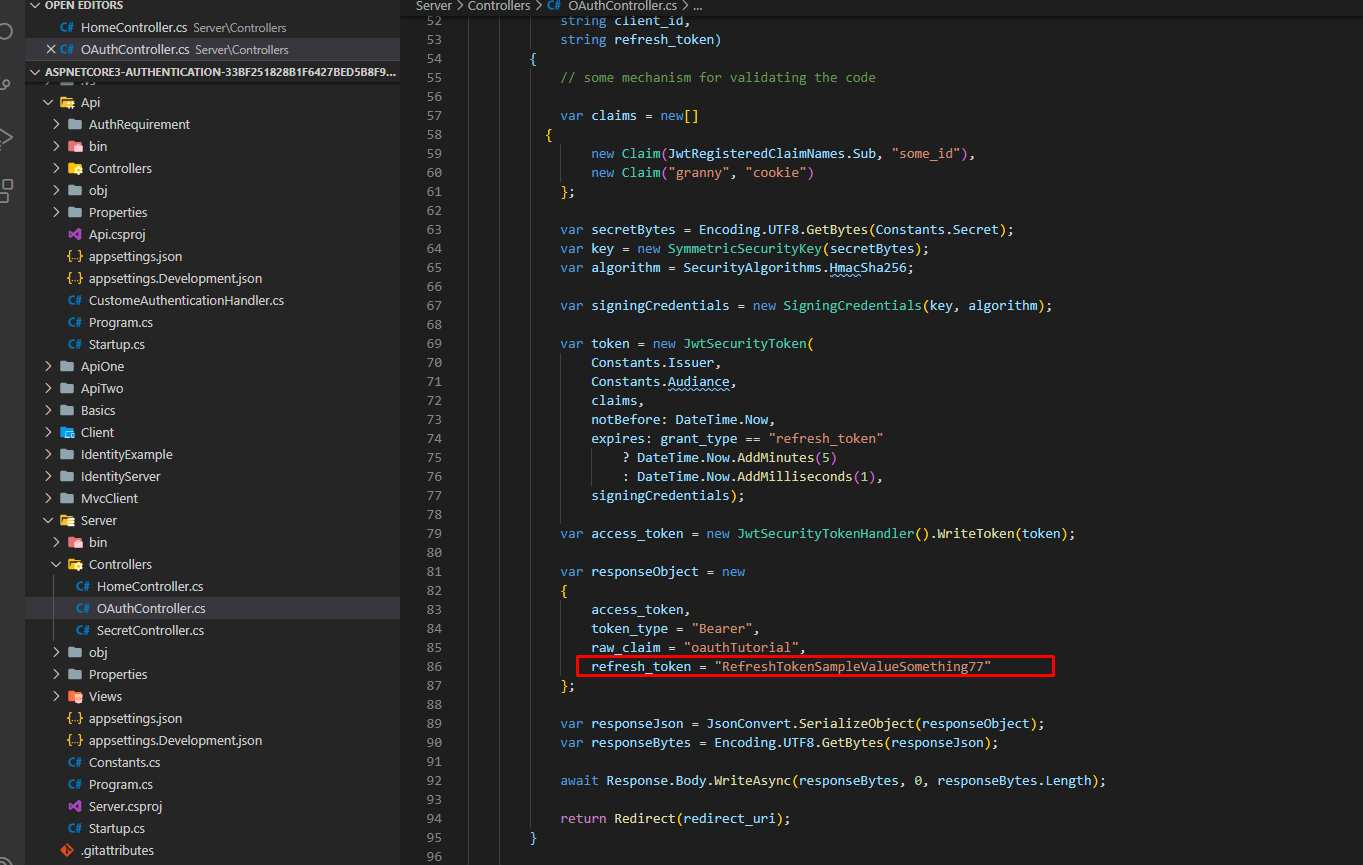


Dùng fresher token thì grant\_type phải là refresh\_token



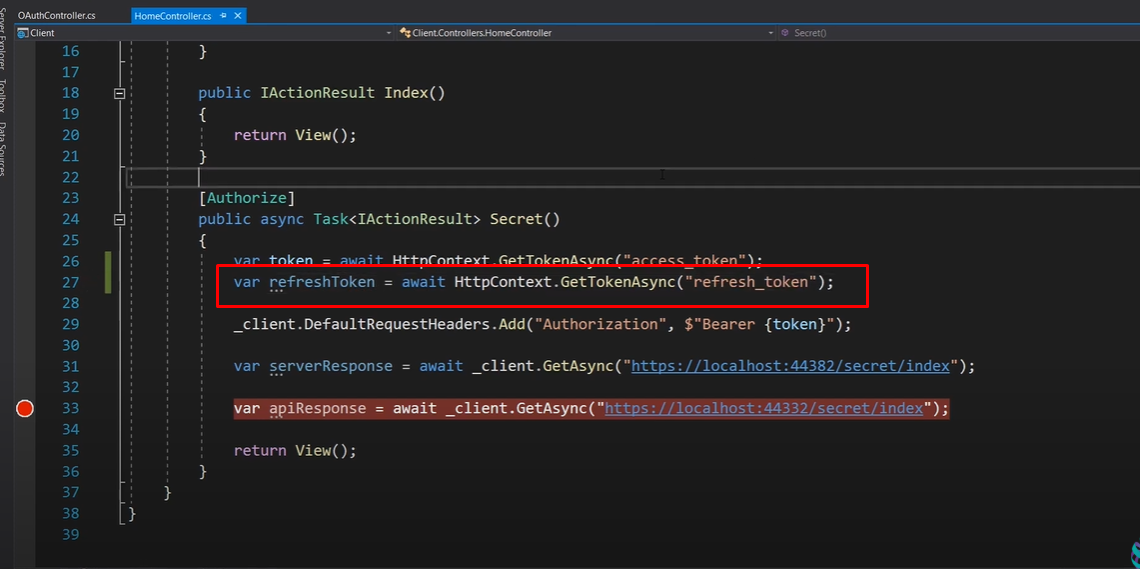
ở đây nghĩ là trường hợp refresh token thì sẽ expire trong 5 phút

còn những trường hợp token còn lại như access token thì expire trong 1 milisecond



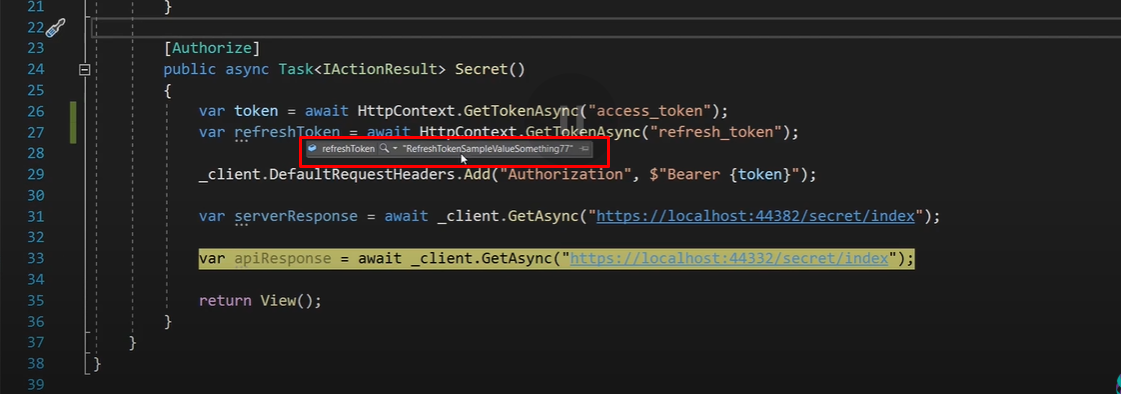
Đưa refresher token vào khi trả về cho client

Việc cấu hình cho refreshe token đã được đưa vào access token rồi, còn refresher token có thể là string nên đưa đại vào cũng được, expire time của refresh token đã được cấu hình ở trên (ảnh trước chổ tạo access token)

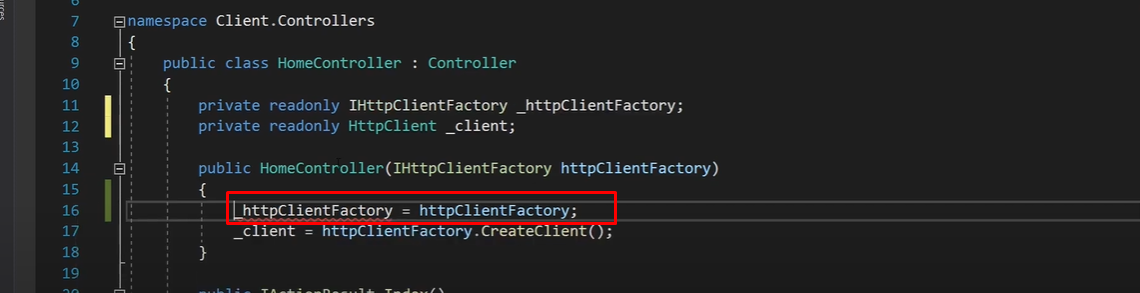


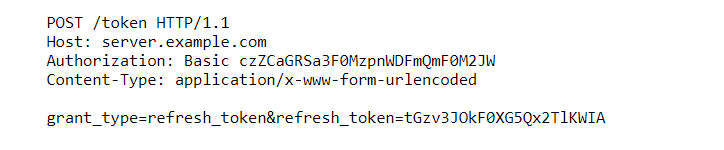
Có thể lấy ra refrehse token

ở đây như là sử dụng client, hoặc postman để test, nhưng mà ở đây là sử dụng code

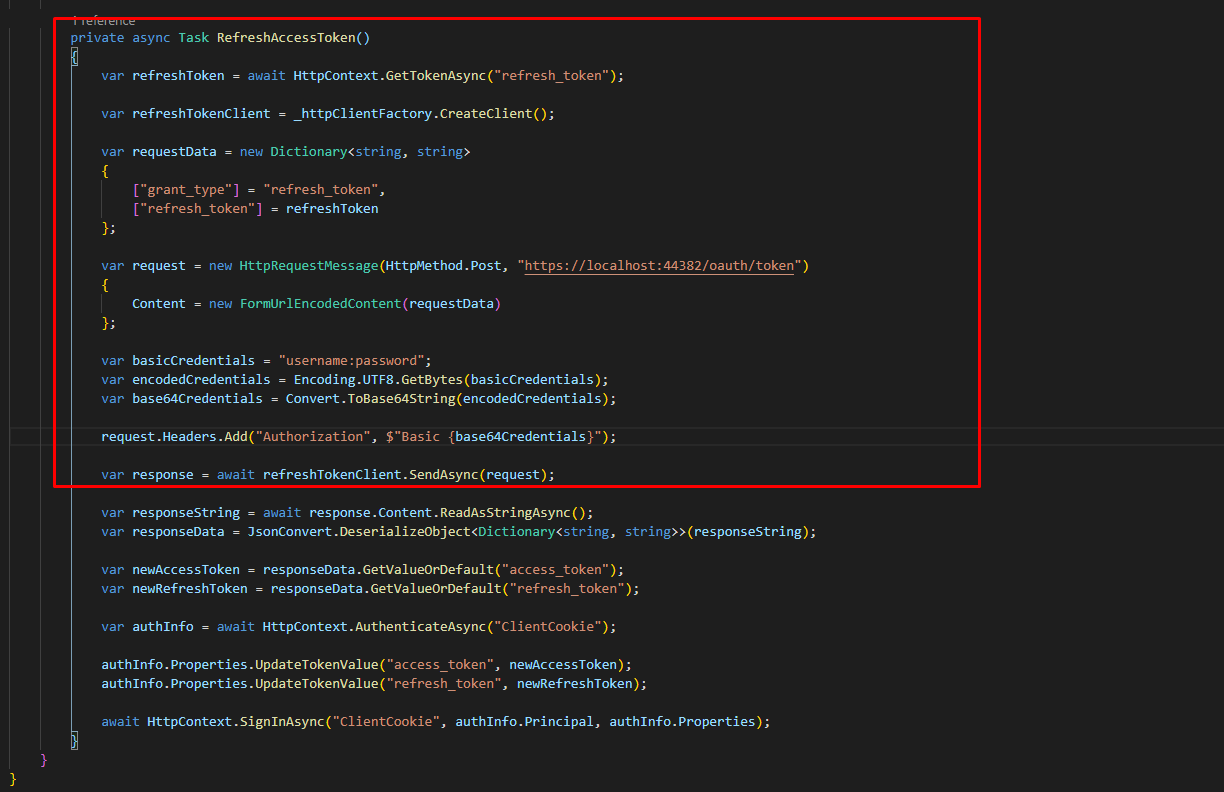


Refresh toke đã được trả về

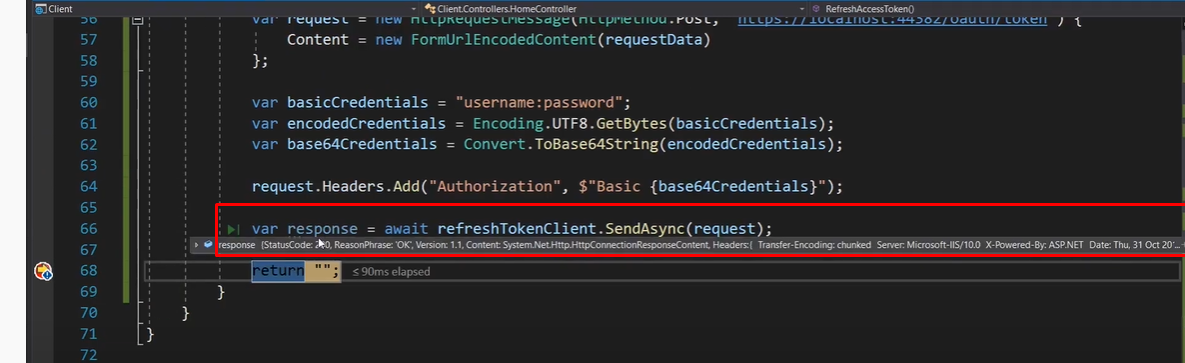




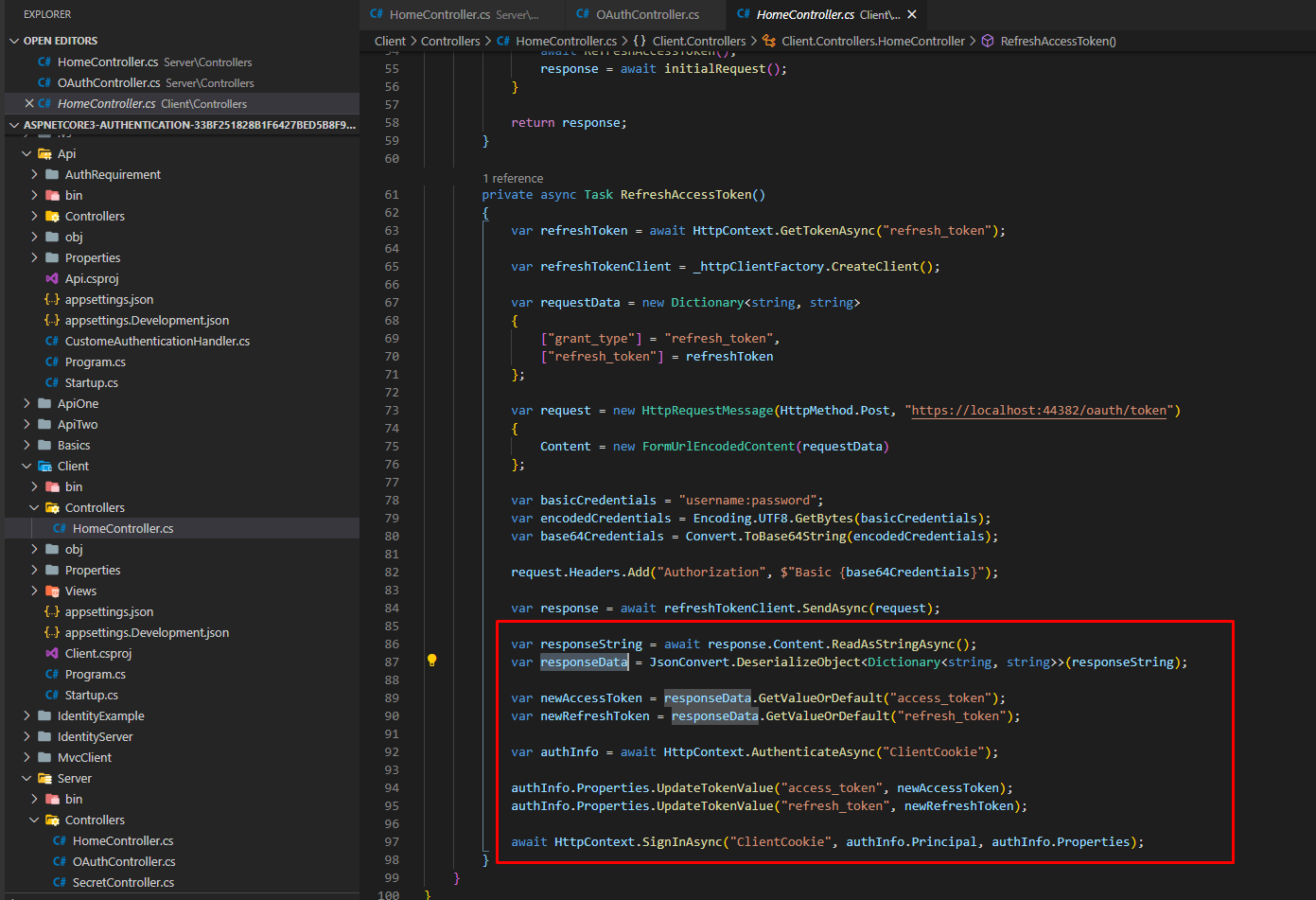
Một request sẽ có dạng như sau



Implement request như vậy



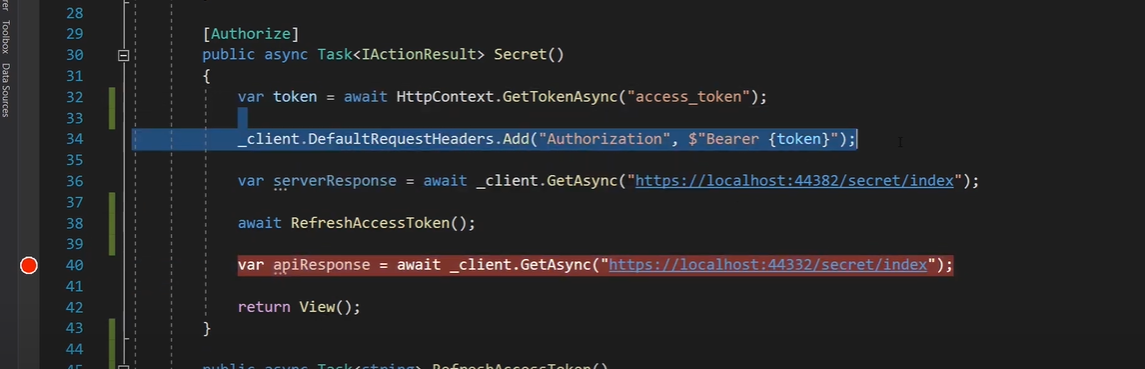
Đã có response



Sau khi nhận được response thì json conver respone đó ra dictionary (như lưu kiểu key, value)

Sau đó lấy access\_token và refresh token, rooif update token này vào authInfo rồi, gọi hàm signInAsync để update giá trị vào cookie

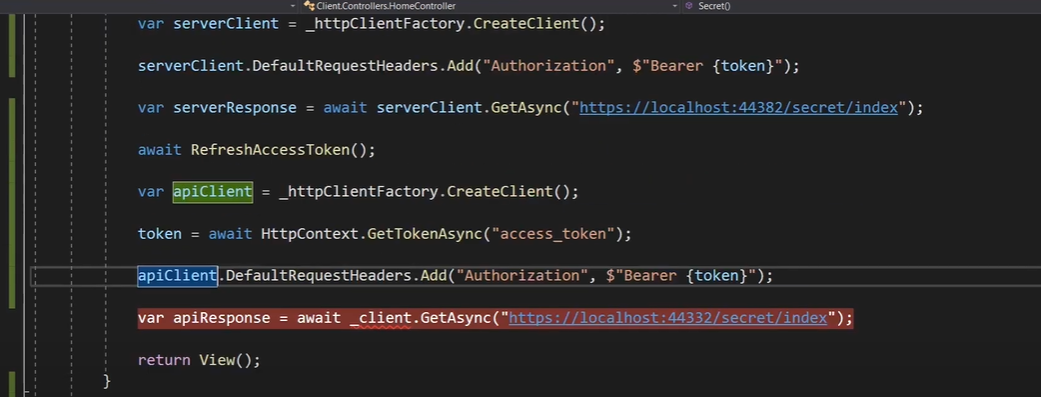
authInfo là nơi token được lưu, từ authInfo có thể acess tới các thành phần auth được lư, và thay đổi



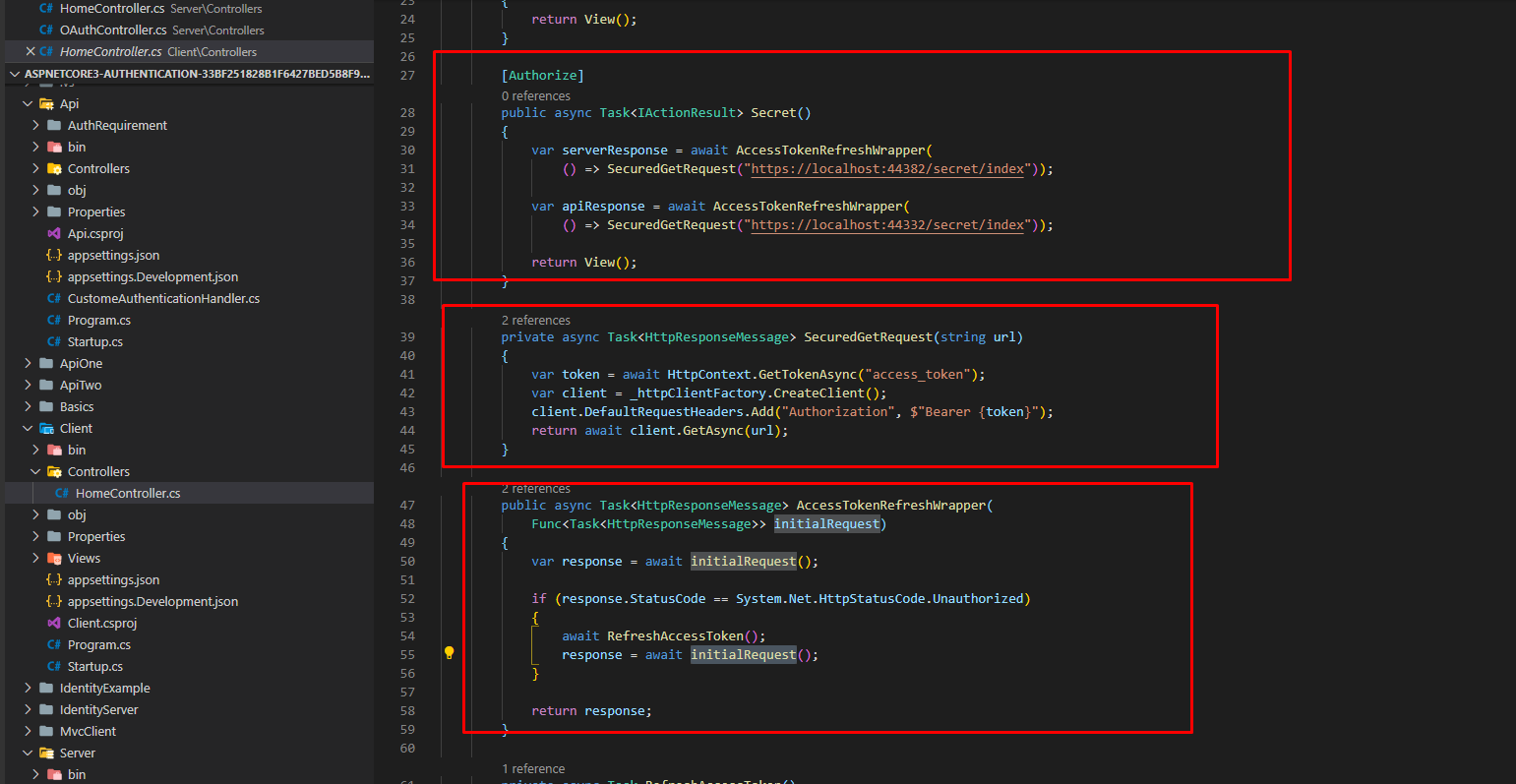
Lấy token ra, nhưng khi request mà ko có hàm refresher token thì nhận lỗi 401, vì token hết hạn

Gọi hàm refresher token thì có thể access vì lúc này token đã được refresh

Trong thực tế có thể check nếu hết hạn thì refresh token chả hạn (ý kiến cá nhân)



Sau khi thay đổi access token thì request lại với token mới



Viết lại hàm như sau

Hàm AccessTokenRefreshWrapper có tham số là func, và hàm func sẽ được gọi để lấy response, hàm func này được truyền vào là lambda express của hàm SecuredGetRequest với request url là tham số

AccessTokenRefreshWrapper

Request và check response, nếu response là unauthorized thì gọi hàm refreshe token sau đó request lại bằng cách gọi hàm initial, hma finitial này sẽ truyền vào SecuredGetRequest để thực hiện request

Thay vì gọi trực tiếp hàm thì ta có thể implement theo cách truyền hàm vào như một thông số như trên, khi đó thì ta có thể truyền trực tiếp tham số vào hàm con lúc gọi hàm chính luôn, chứ không cần phải truyền vào hàm chính rồi sau đó hàm con lấy tham số của hàm chính để truyền vào