Mục tiêu

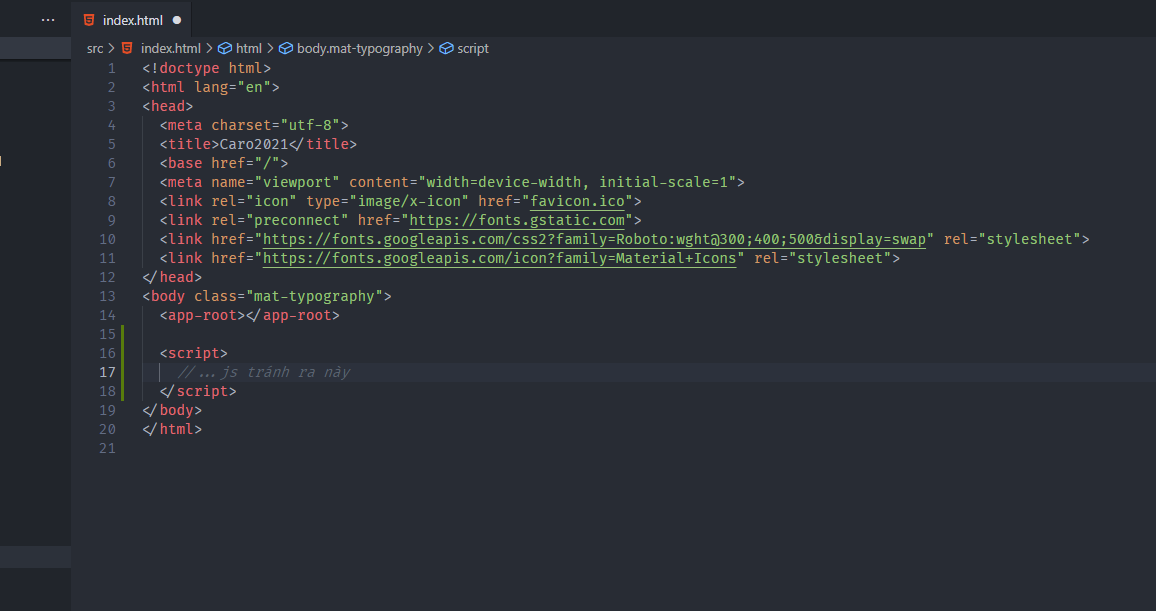
* Xử lý realtime cho angular app
* Xử lý chat ( API, UI)
* Nếu nhanh xử lý nước đi.

Kiến thức: SignalR – đối với back-end, front-end ( Websocket: vì nó xử lý bằng manual- xử lí tay)

UI: kết hợp giữa component trong material, html, css thuần, flex-layout (không sử dụng bootrap)

Cố gắng viết thuần html, css, material angular và xử lý bằng ts là chính.

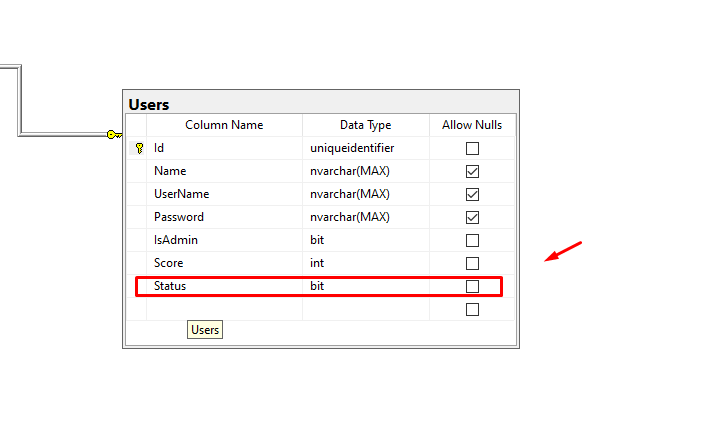
Tránh file js đc nhúng ở trong file index.html



Cần có 1 API lấy ra danh sách người chơi ( bên trong có status chỉ ra rằng người đó đang onl hay k)



Dựa vào status này

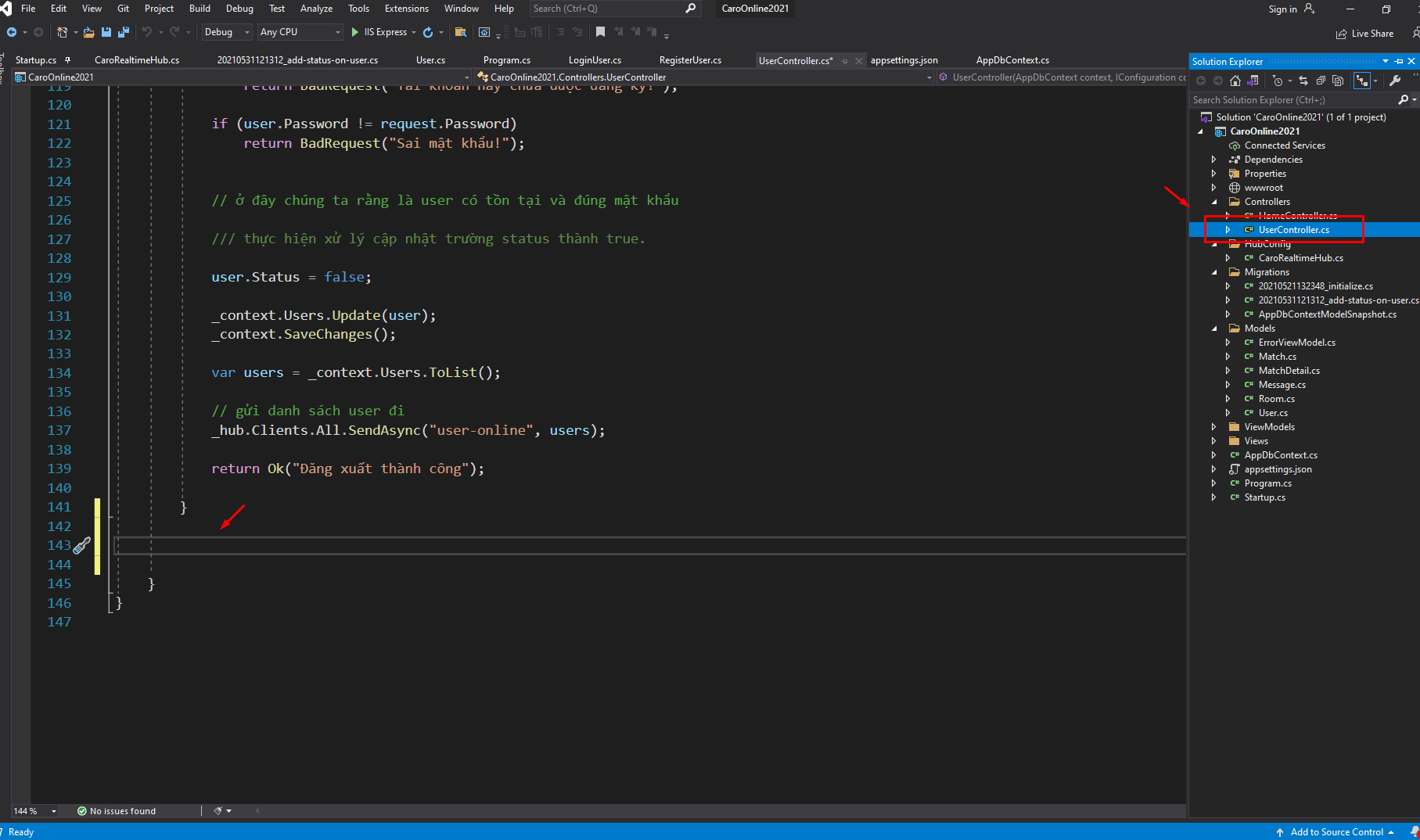


API get-users

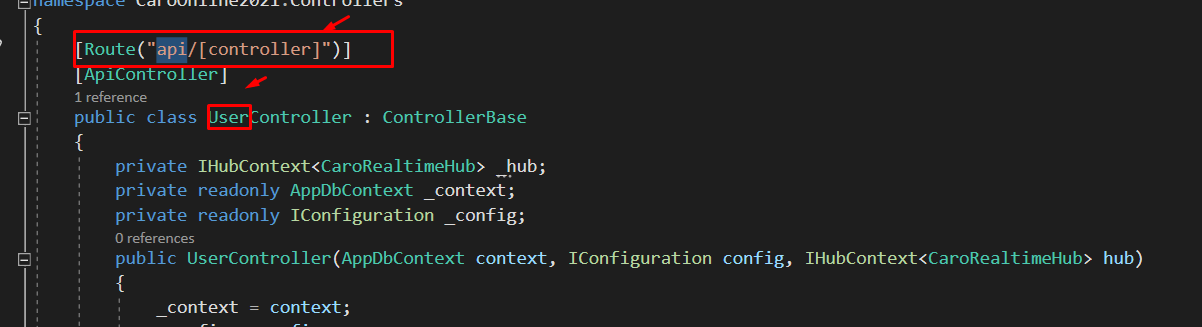
Method: GET

url: <domain>/api/<controller>/get-users => <https://localhost:44313/api/User/get-users>

mục đích là lấy toàn bộ users (kèm trạng thái)



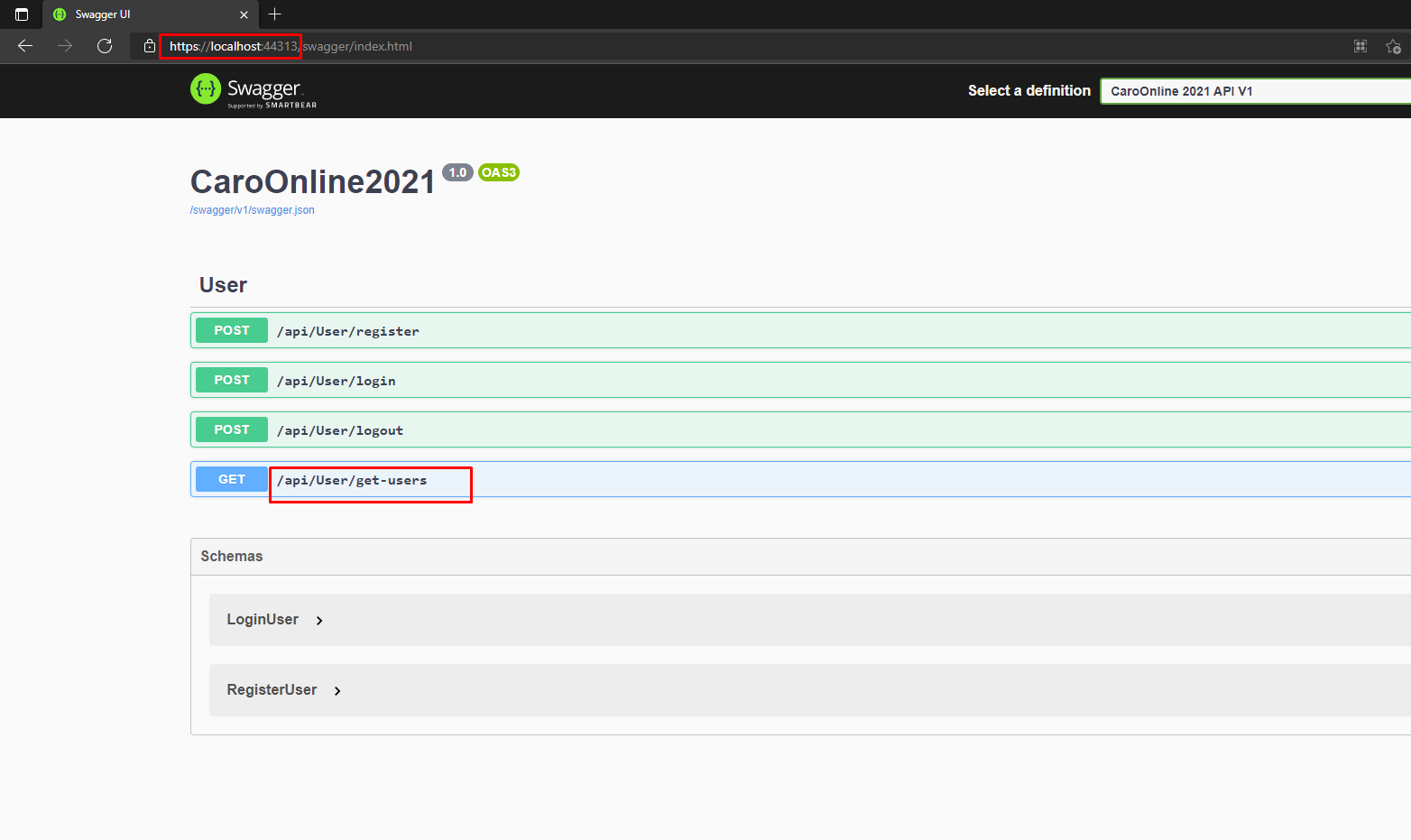
Thường đặt api mới vào vị trí cuối của danh sách api trong controller tương ứng





Tổng hợp 3 thằng này chúng ta ra cái URL

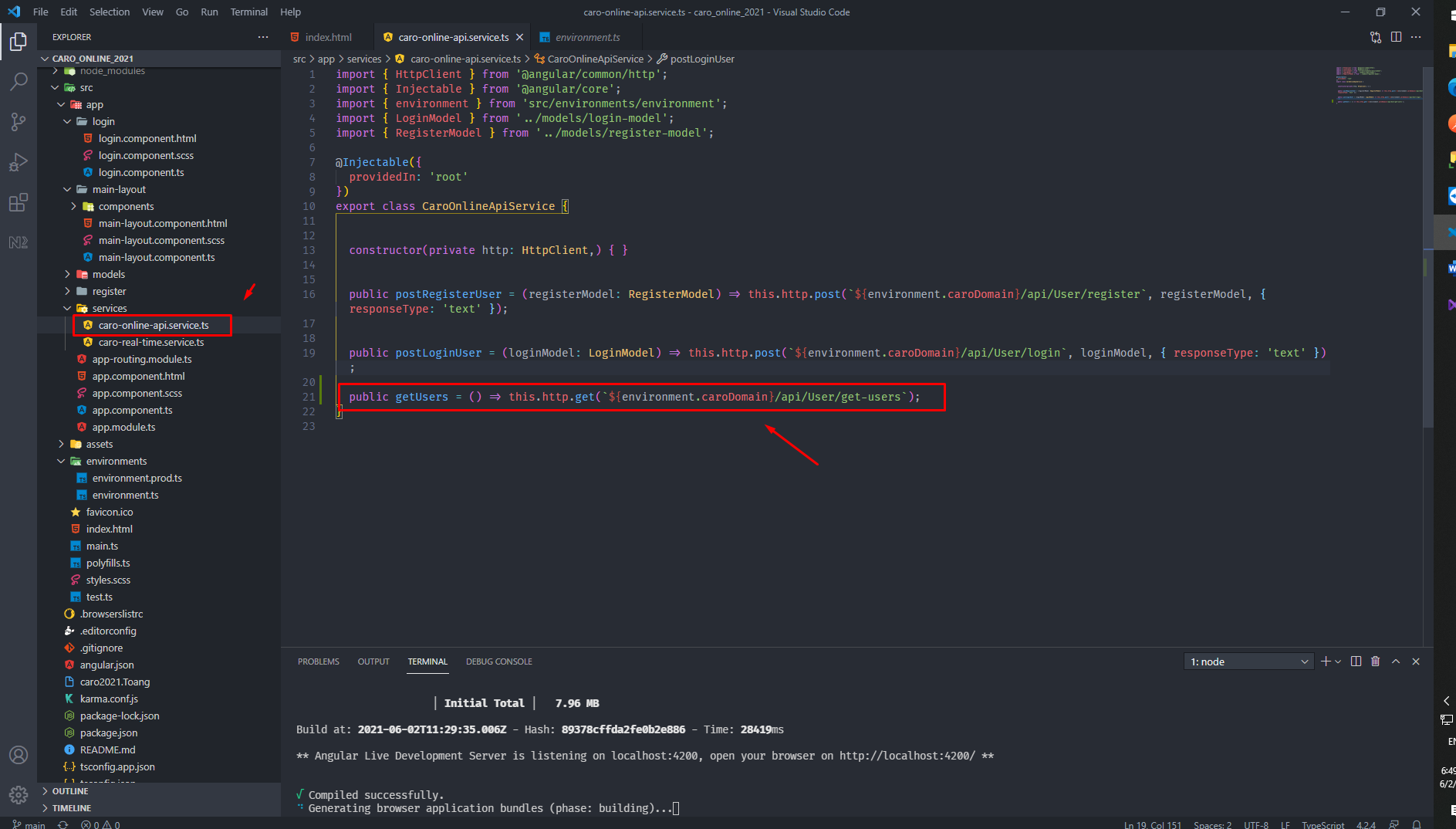
<https://localhost:44313/api/User/get-users>



Như vậy, chúng ta hoàn thành api get toàn bộ users mà có chứa status bên trong

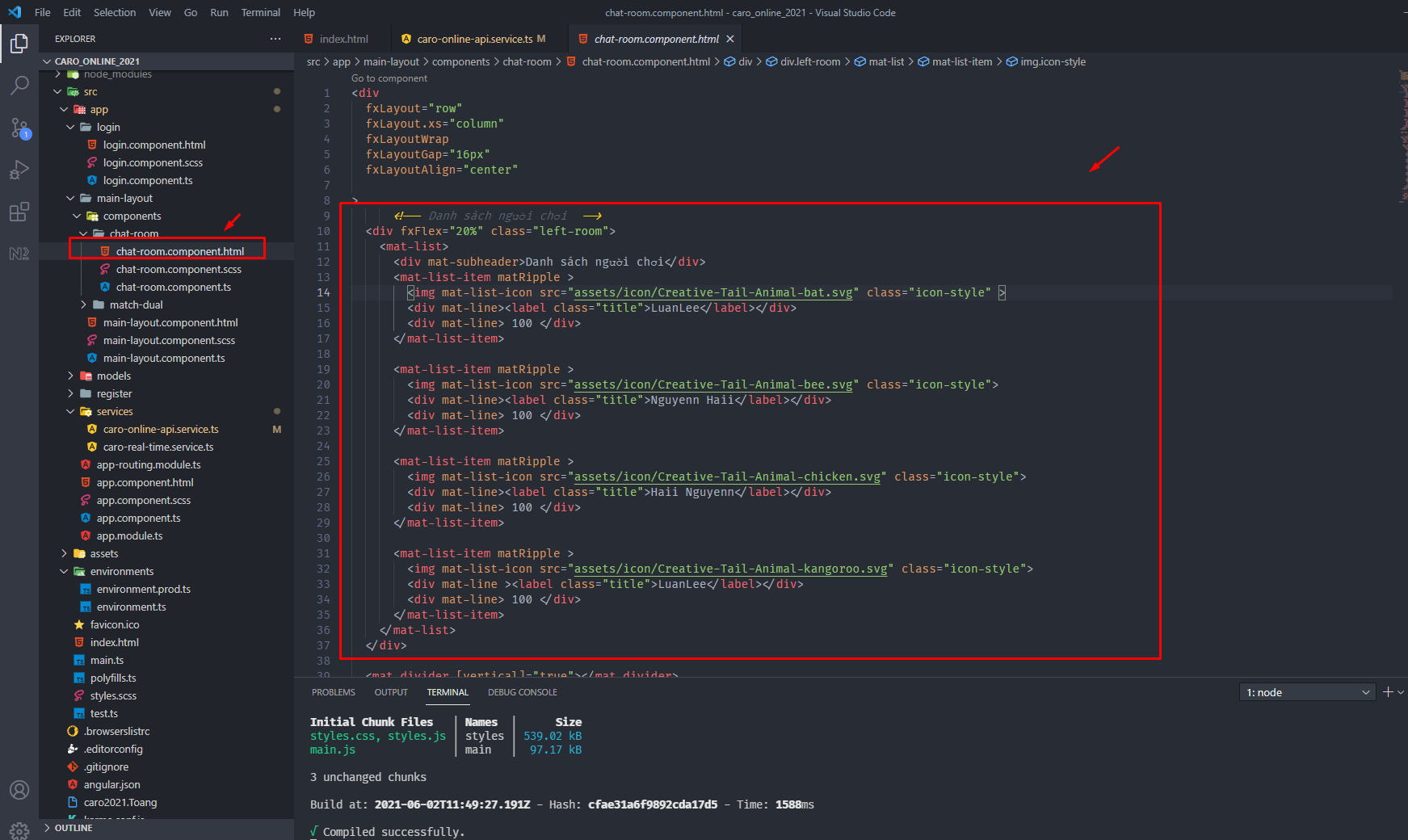
Tiếp theo thực hiện khai báo api này trong front-end (angular-app)

Khai báo api mới get-users vào caonline-services ở angular app



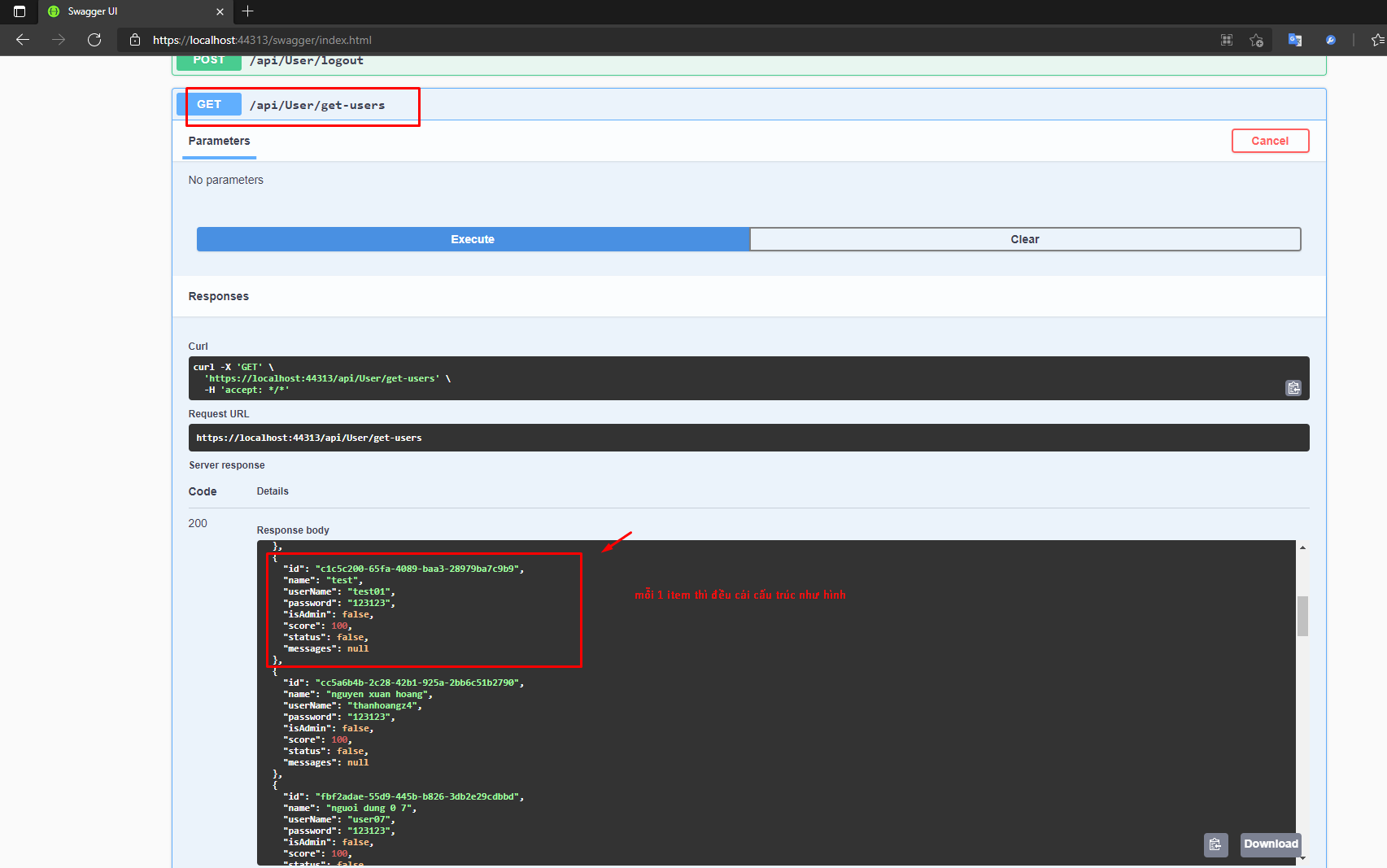
Lúc này chúng ta được quyền sử dụng

Chúng ta tìm tới component mà chịu trách nhiệm hiển thị ra danh sách người chơi ở đâu đó.



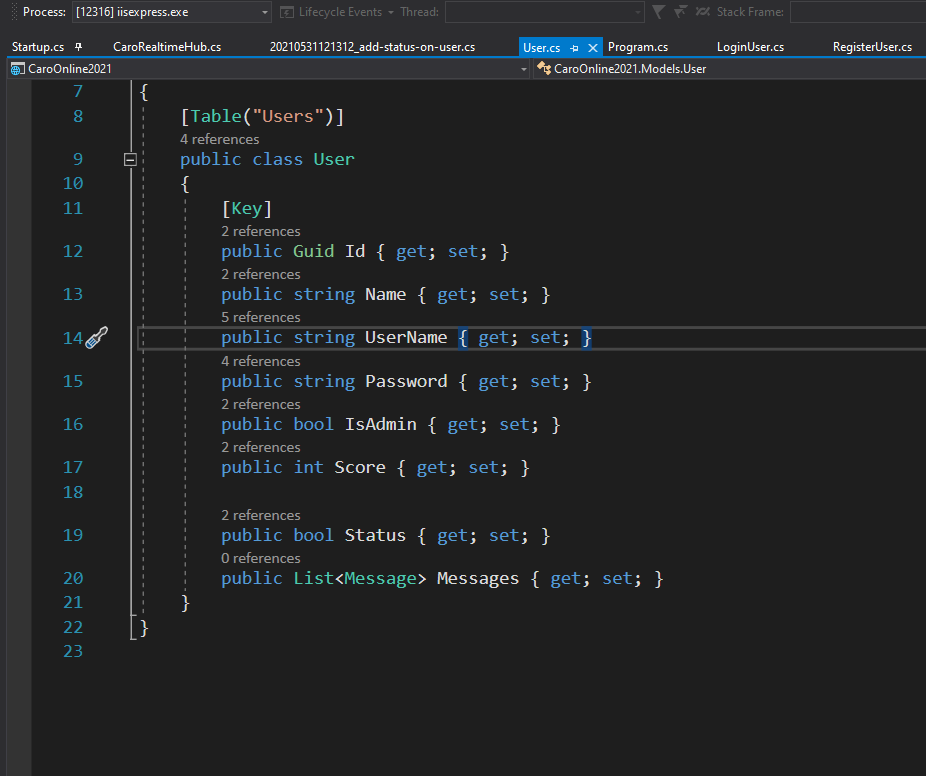
ở ví dụ tìm thấy ở chat-room

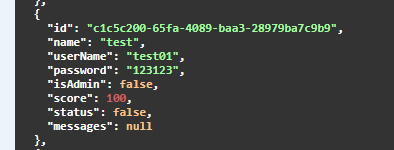
ý tưởng: bây giờ cần 1 biến là 1 mảng object có các thuộc tính là giống với kiểu liệu trong api trả về



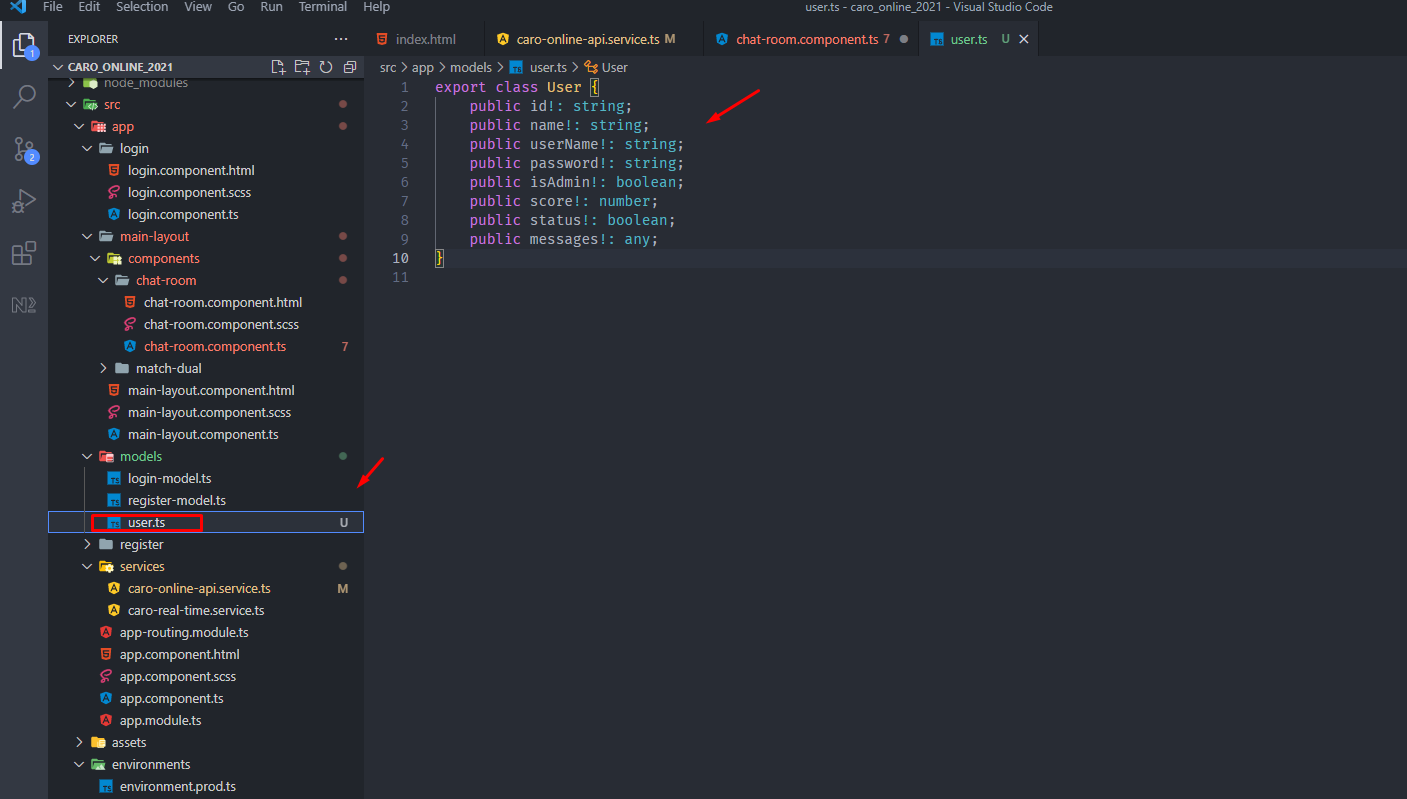
Tạo ra 1 class có các thuộc tính như hình trên

Lưu ý class này giống hệt class model ở back-end nhưng khác ở chỗ bị down ( lower cái kí tự đầu tiên)





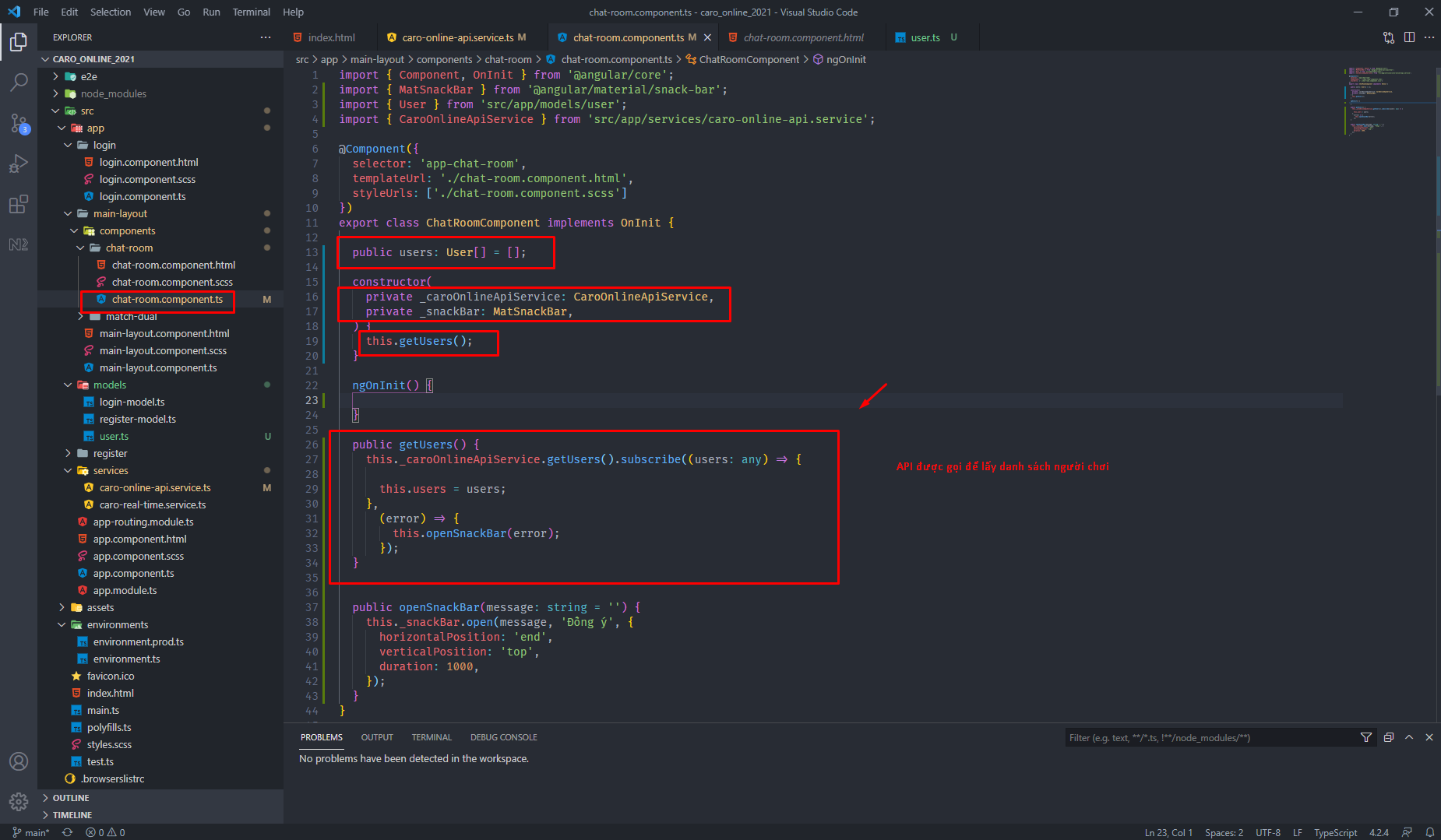
Do đó muốn bắt chính xác thì phải định nghĩa class theo kiểu dữ liệu respond ( tức là theo cái class bị lower kí tự đầu tiên)



Tạo model – class đặt tên user => tên class là User

Sau đó viết func call api users

Trong angular



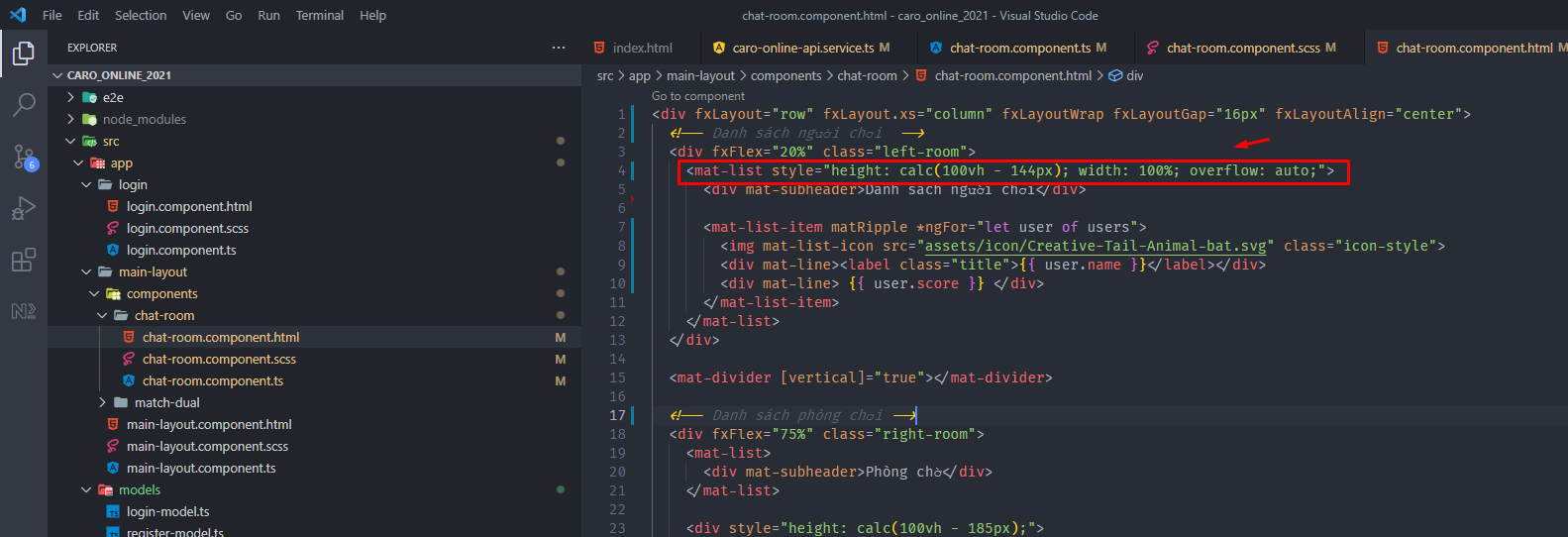
Lưu ý: api này gọi thành công khi status code = 200 thì nó nhảy vào subscribe tức là gắn danh sách users từ back-end trả về cho biến users (mảng rỗng)

Mục tiêu là binding biến users ra html ( 1 chiều từ file ts => html)

( sử dụng dấu {{ <giá trị>}} )

Vì nó 1 mảng cần hiển thị nên lập tức chúng ta nghĩ ngay đến câu lệnh để lặp qua các phần tử từ ts => html

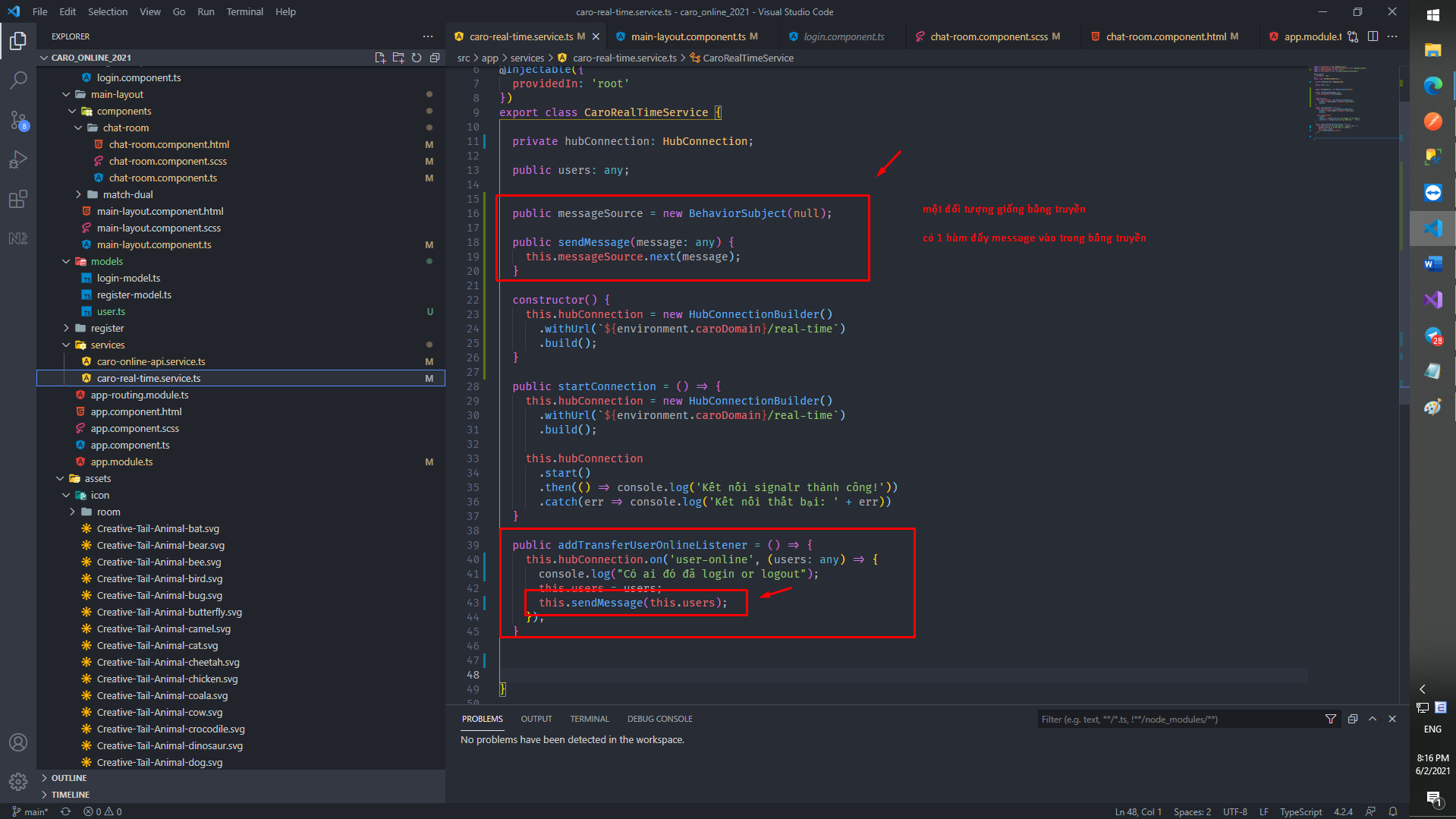
( sử dụng ngFor cho việc lặp qua các item)



Sử dụng ngfor và {{}} one way binding cho file html

(Bài tập – random cái icon trong file asset)

Tiếp tục phần xử lý realtime để báo rằng user nào đang online hoặc offline ( đồng thời)



Server realtime dùng chung sẽ lắng nghe từ server ( dòng 40 – 44 )

ở dòng 43: gửi thông điệp (danh sách người dùng mới có status đã được cập nhật) đưa vào messageSource( hiểu là băng truyền)

