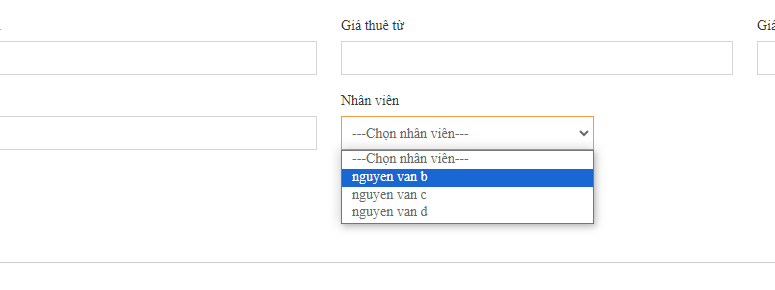
Buổi 24:

Giờ ta sẽ đổ data từ DB lên các thẻ seclect theo status nếu 0 thì sẽ không lấy nếu là 1 thì sẽ lấy (việc ta thêm một điều kiện nhỏ này có thể thay đổi nếu nhân viên đó không hoạt động)

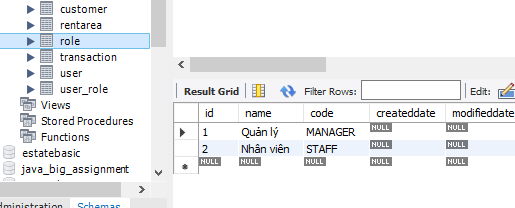


Ý tưởng ở đây như sau : khi mà nói đến lấy dữ liệu dưới DB ta nghĩ ngay đến mô hình 3 MVC và trường hợp này cũng không ngoại lệ, việc lấy data từ các nhân viên liên quan đến bảng user và role nên ta cũng sẽ đi từ UserSevice rồi đến UserRepository.

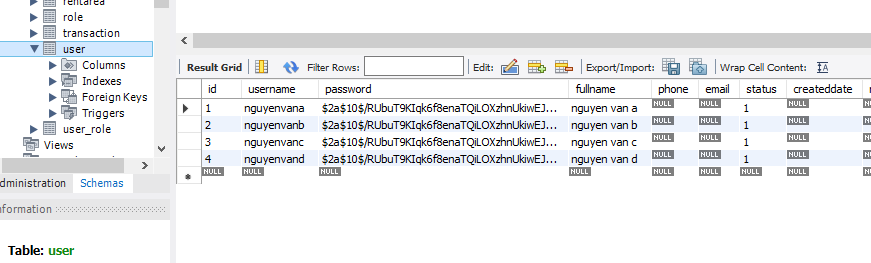
Giờ cùng nhìn dưới Db:

Ta sẽ lấy những user nào nào có role code là staff:

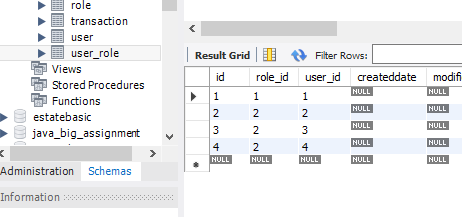
-ở bảng role ta thấy id-role= 2 sẽ là STAFF

****

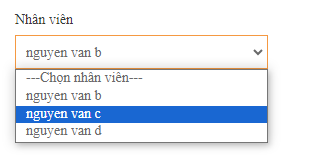
Đây là bảng user và tất cả các bản ghi đều có status đều là 1



Bảng user-role thì có 3 ông user có role là staff là id=2,3,4 và quay lại nhìn thì status của chúng đều bằng 1 , vậy 3 thằng này sẽ được hiện thị ra.

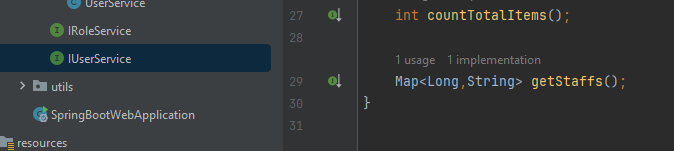


Quay trở về select sau:

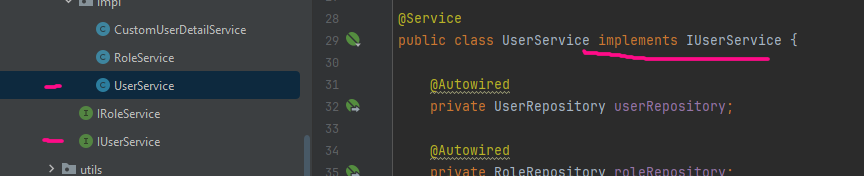


Mục đích khi ta chọn 1 option mà 1 option đó sẽ chứa 2 thứ :thứ nhất id là thứ mà ta sẽ gửi data để tìm kiếm (đó chính là cái thuộc tính value của thẻ option) , thứ 2 sẽ là tên của nhân viên (chính là label mà ta hiển thị) , vậy nên để lấy data dưới DB hiển thị lên view ta chỉ cần map để chứa những data thỏa mãn rồi sau đó dùng modalandview để xử lí

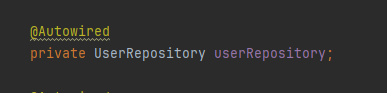
Đầu tiên ở tầng UserService ta sẽ thêm một phương thức ảo trả về một map.



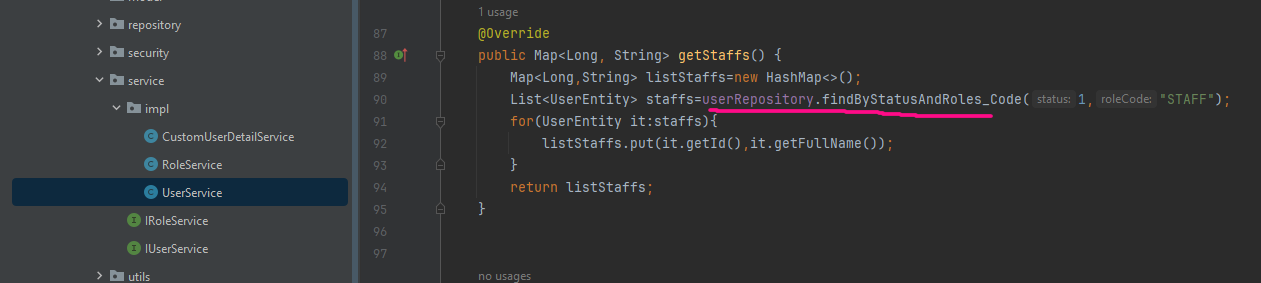
Triển khai hàm đó:



Nhớ:

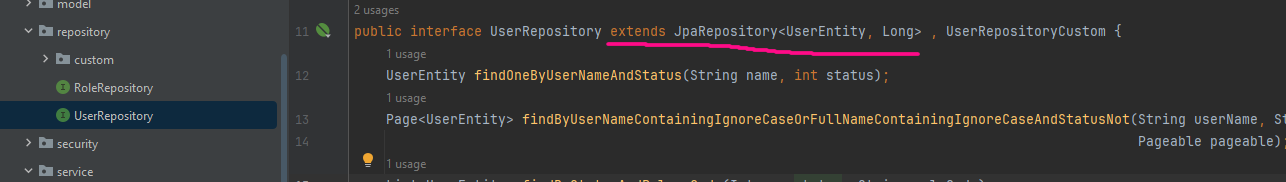


Ta sử dụng hàm xây dựng sẵn của Spring data JPA để tìm trả về toàn bộ các nhân viên có role là staff và có status là 1 thông qua phương thức .findByStatusAndRoles\_Code(1,"STAFF");

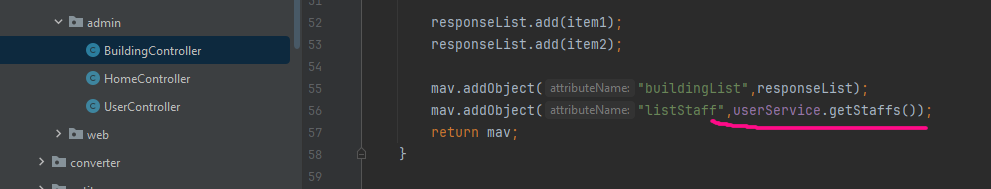


Sau đó là bước xử lí để trả về data.

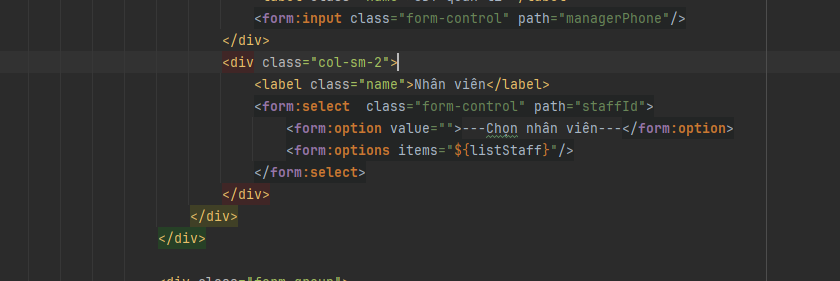
Hãy nhìn qua tầng Repo , (nhắc lại hãy implement JpaRepository<Entity,DataType> để xử dụng springdatajpa)



Quay về controller:

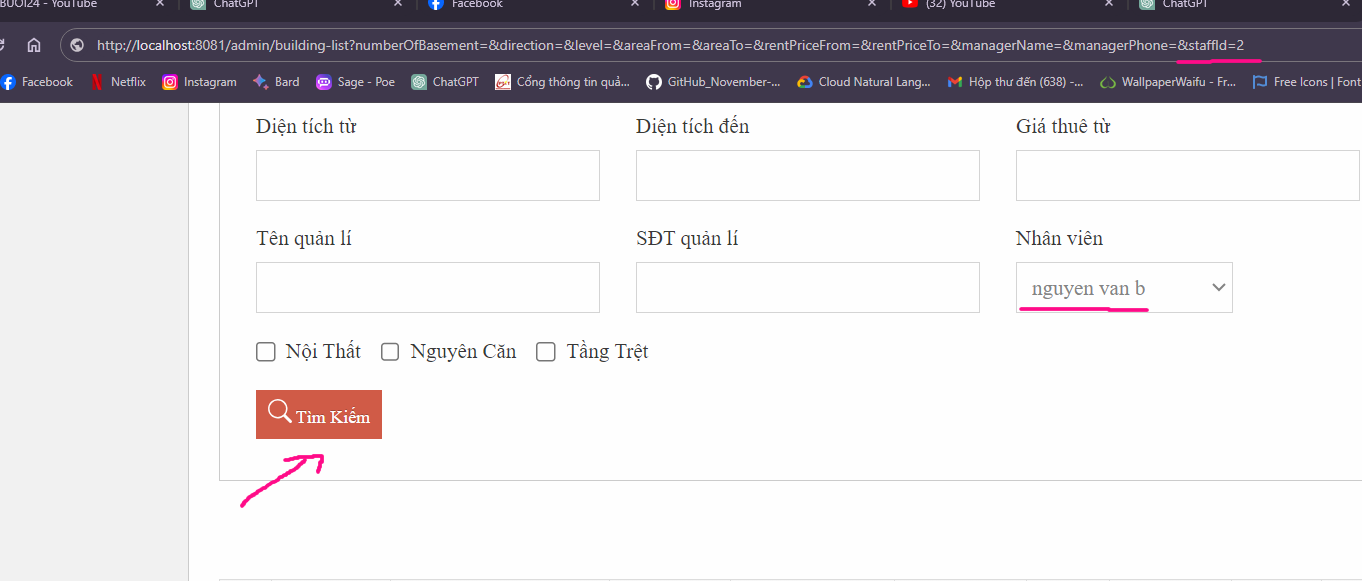


Sang bên giao diện:



Ta ngoài tiêu đề dùng thẻ <form:option> ra, thì những option sẽ sử dụng thẻ <form:options> phương thức items sẽ chứa list đối tượng và mỗi một đối tượng là cặp key-value theo dạng map, key ở đây vì nó không trùng nhau vậy nên ở tầng service tôi đã để cho id của user chính là key của map và thằng item này sẽ gán luôn cái key đó cho value của thẻ option(  ) và thằng value (hay là tên user ) của map sẽ được gán cho label của thẻ option.

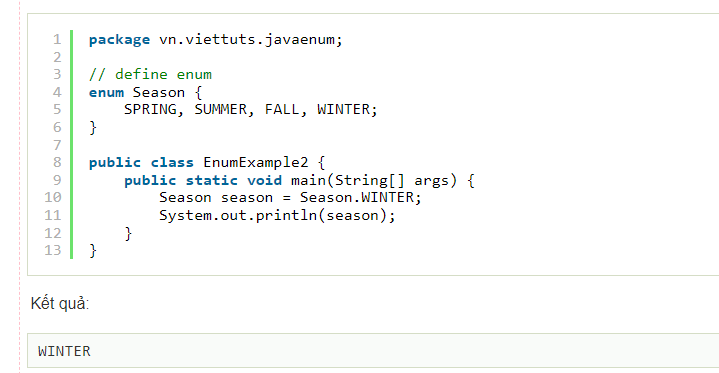
Hãy nhìn lên param sau khi tôi chọn thằng b và click nút tìm kiếm.



JAVA ENUM.

**Enum trong java** là một kiểu dữ liệu đặc biệt của Java được sử dụng để định nghĩa các tập hợp các hằng số. Cụ thể hơn, Java enum là một **kiểu đặc biệt** của lớp trong java. Một enum có thể chứa các trường, phương thức và Constructor.

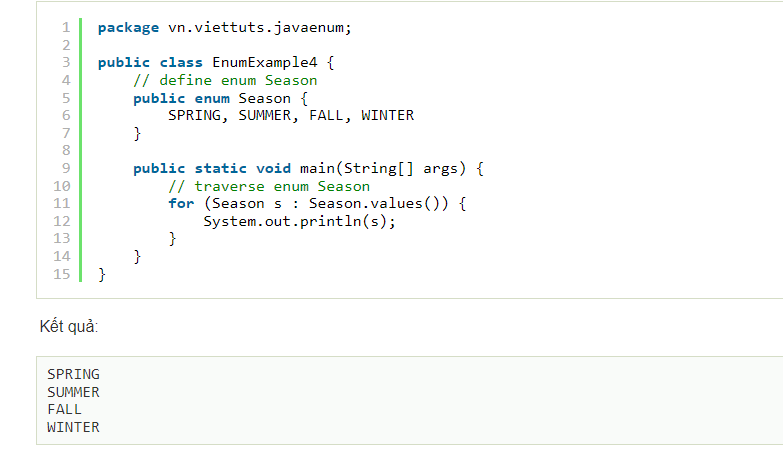
-Enum trong java có thể được định nghĩa bên trong hoặc bên ngoài một lớp, vì nó tương tự như class trong java, hoặc định nghĩa ở 1 file riêng biệt.



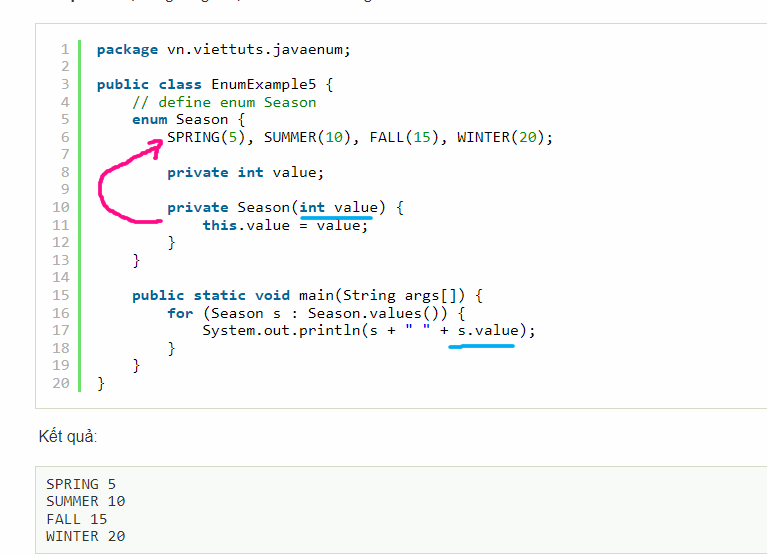
Phương thức values() trong enum

Trình biên dịch trong java tự động thêm phương thức values() vào enum khi nó được biên dịch. Phương thức values() trả về một mảng chứa tất cả các giá trị của enum.

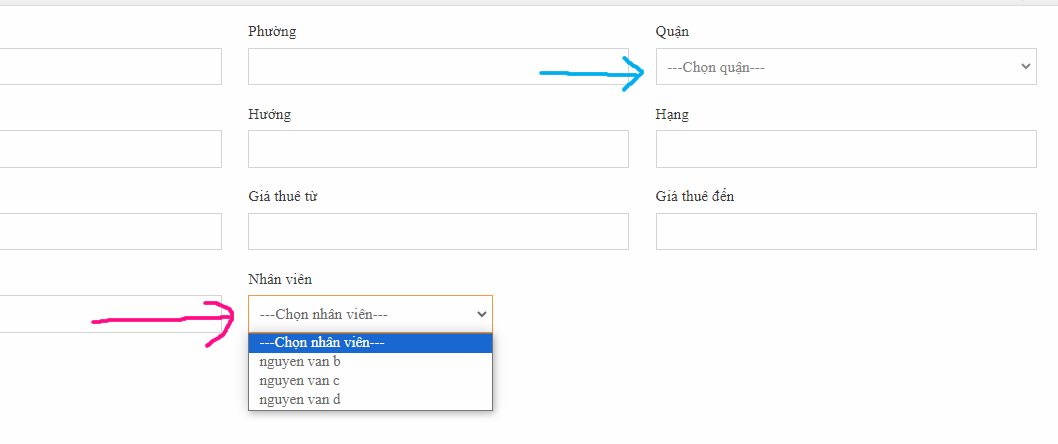
Duyệt các phần tử trong enum:



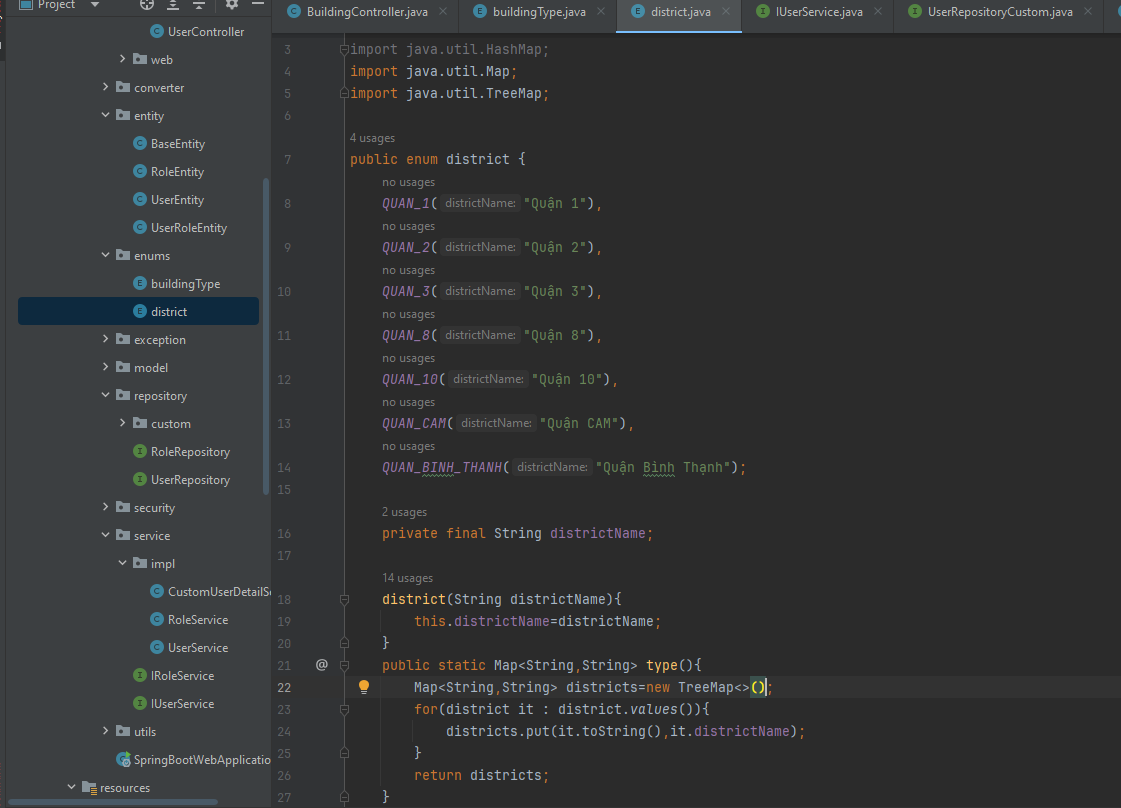
chúng ta có thể khởi tạo giá trị cụ thể cho các hằng số enum bằng cách định nghĩa các trường và các Constructor.



Ok chúng ta đã đi qua về enum vậy thì thằng này sẽ có ích gì giúp ta trong project:

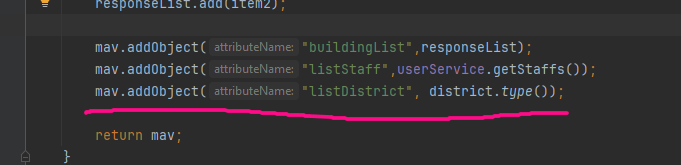


Nhìn các filed trên thì ta thấy có filed nhân viên và quân là dạng option , chúng ta đã từng xử lí thằng quận thông qua việc sử dụng mô hình 3 mvc , thằng quận cũng có thể làm tương tự nhưng ta sẽ không làm vậy mà ta sẽ sử dụng enum bởi trong thực tế nhân viên thì có thể thay đổi thường xuyên cho nên việc sử dụng đến DB là không tránh khỏi , nhưng Quận thì rất lâu mới có thể thay đổi , cứ thử nghĩ xem ? vậy nên ta sẽ sử dụng enum cho những thằng ít bị thay đổi.

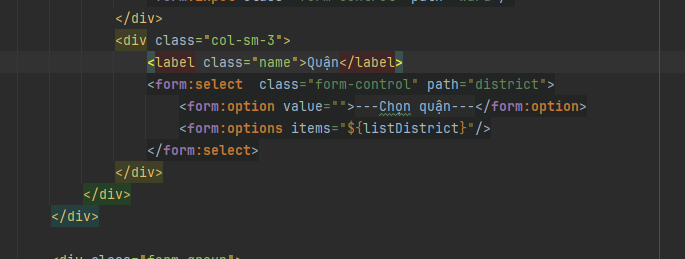


Xây dựng enum xong giờ công việc xử lí ra view rất giống với thằng User

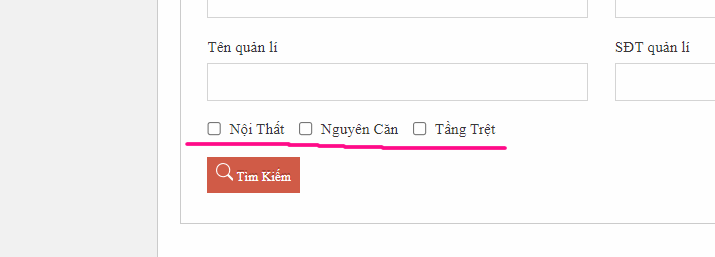
Quay về controller



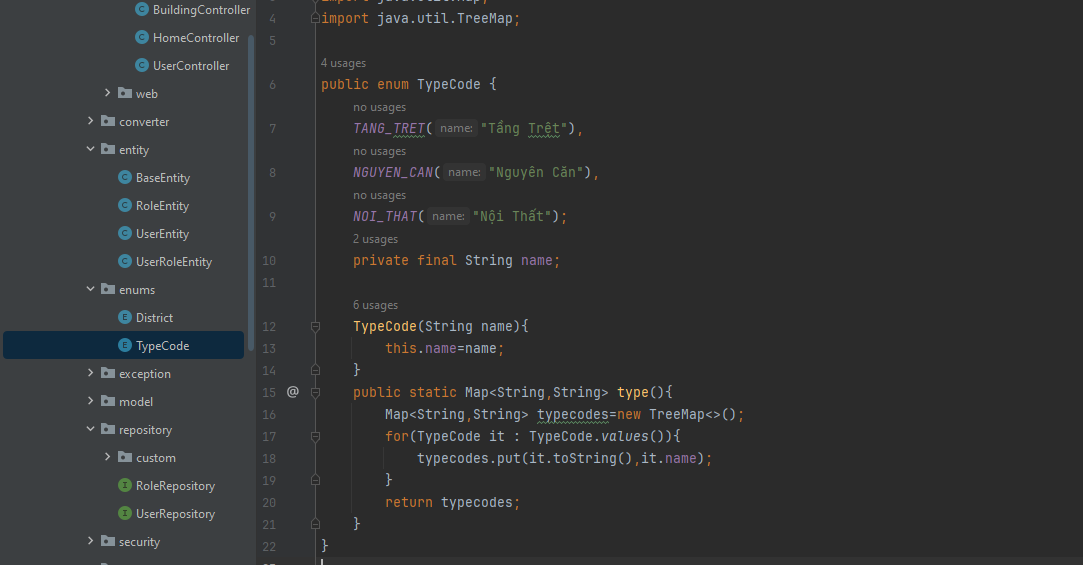
Ra view :



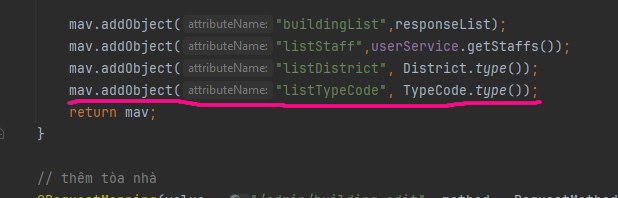
Tiếp theo ta sẽ cần sử dụng đên enum để sử lí thằng sau :



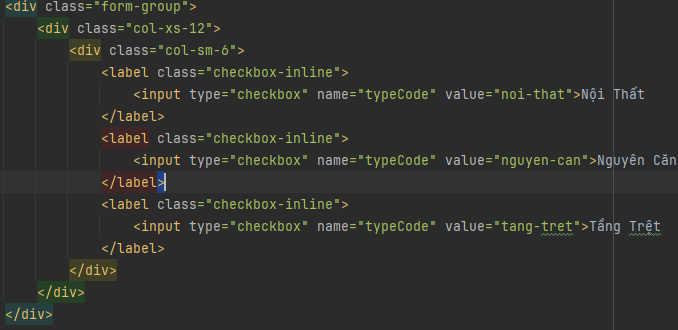
Tại sao lại sử dụng enum , tại vì bạn có thấy khi 1 tòa nhà xây nên có ai lại đi thay đổi kiểu cách của nó không vậy nên nó rất ít bị đổi => dùng enum



Quay ra controller:



Ra view ta sẽ không sử dụng thẻ input có type =”checkbox”



Sử dụng thằng sau:

