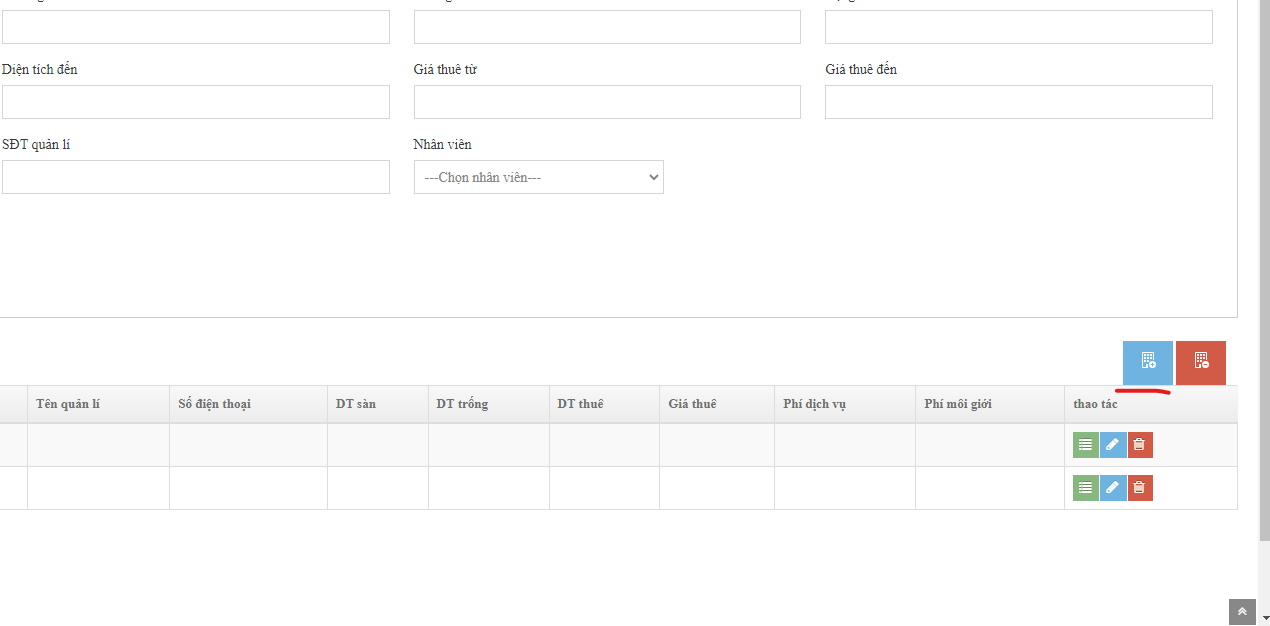
Buổi 22: thác mắc không hiểu chạy server ra trang admin kiểu j, và tại sao cần database thì mới chạy đc ra.

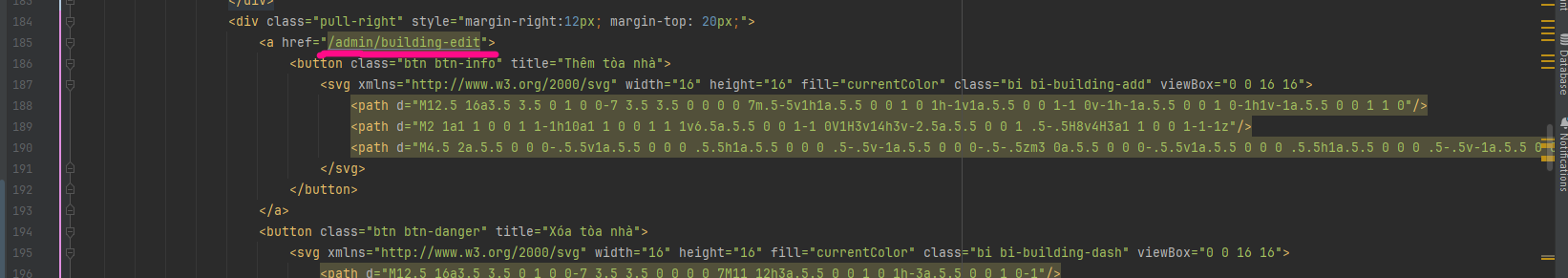
Để nhúng 1 file html vào dự án ta sẽ dùng ModelAndView :

Đầu tiên chúng ta sẽ sử lí sự kiện khi bấm vô một cái gì đó chẳng hạn như 1 nút thì nó sẽ nhảy ra một view khác (html)

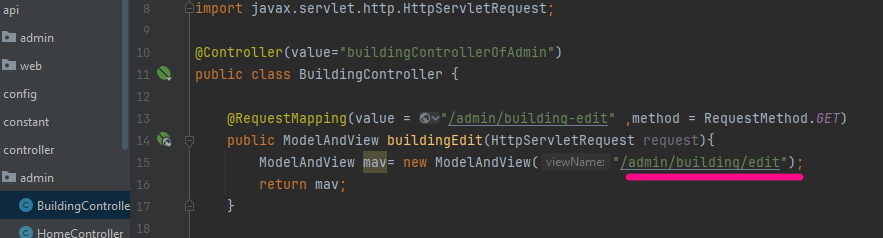
Sau đây tôi sẽ sử lí cái view hiện ra khi bấm vào tên tòa nhà như sau :



Vào phần mã nguồn html của nút đó rồi làm như sau:



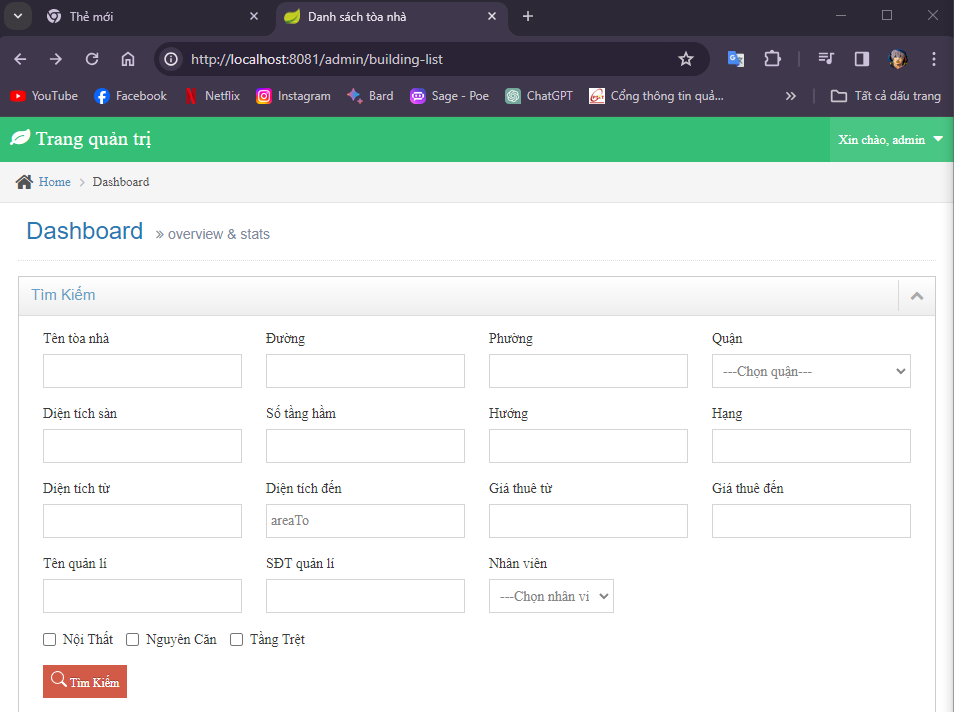
Bạn nhìn thấy ban đầu nó chỉ là một cái nút button từ dòng 186 –đến 192 nhưng ta sẽ bọc nó vào bằng 1 thẻ a mục đích là để khi click vào nó sẽ gọi cái url (api) được để trong href. Thì khi mà gọi đến cái api đó chúng ta sẽ quay sang controller để sử lí tiếp như sau:



Trước khi giải thích tôi sẽ lưu ý đó là http method ở đây bắt buộc phải là GET nhé tại vì hành động chúng ta làm là click và lấy một cái view.

Tạo một cái đối tượng thuộc ModelAndView và truyền cho nó đường dẫn tới filde html đó (minh họa phần gạch hồng) rồi tiếp theo công việc là return về đối tượng đó thôi =))

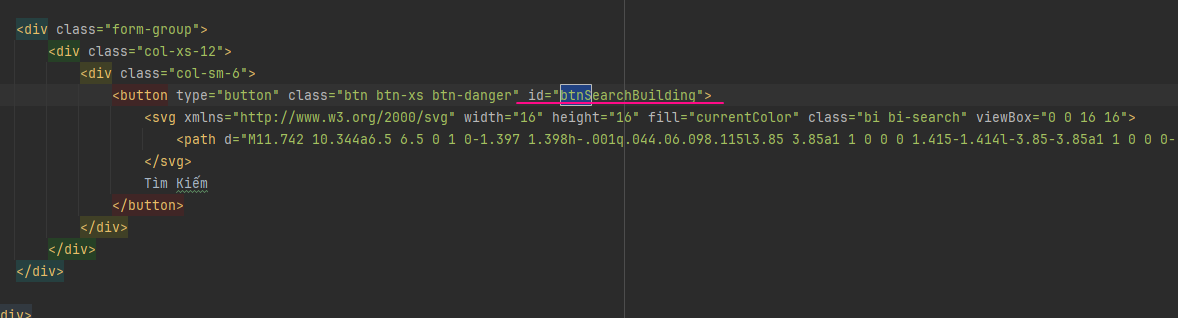
Buổi 23:

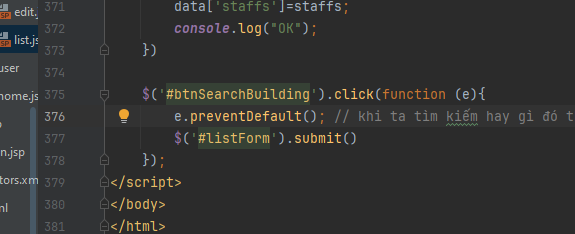


Giờ ta sẽ làm tiếp bước gửi data : bài toán đặt ra ở đây là làm thế nào để ta điền vào các filel rồi bấm tìm kiếm thì các yêu cầu đó sẽ được gửi lên thanh param và gửi về BE.

Ý tưởng ở đây sẽ là: toàn bộ các filed kia bình thường là nó sẽ gói trong 1 thẻ form đúng không, nhưng thay vì dùng thẻ form đó ta sẽ dùng thẻ form:form của spring framework sau đó ta sẽ xử lí sự kiện đối với buttom tìm kiếm.

Ta sẽ đặt id cho button rồi gọi id đó ra rồi xây dựng function cho nó:

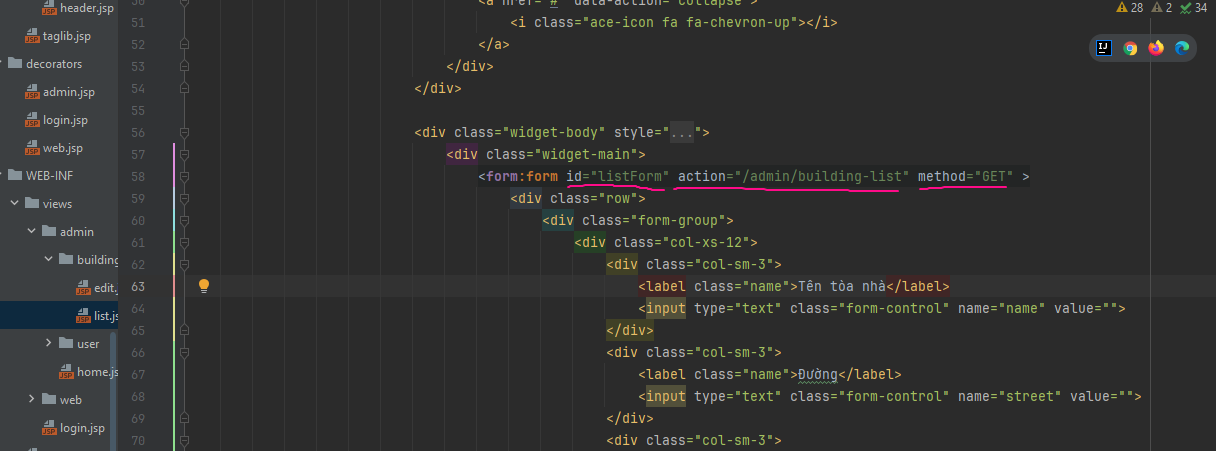




Trong mã JavaScript trên $('#listForm').submit() là một sự kiện gửi (submit event) được kích hoạt trên form có id là "listForm".

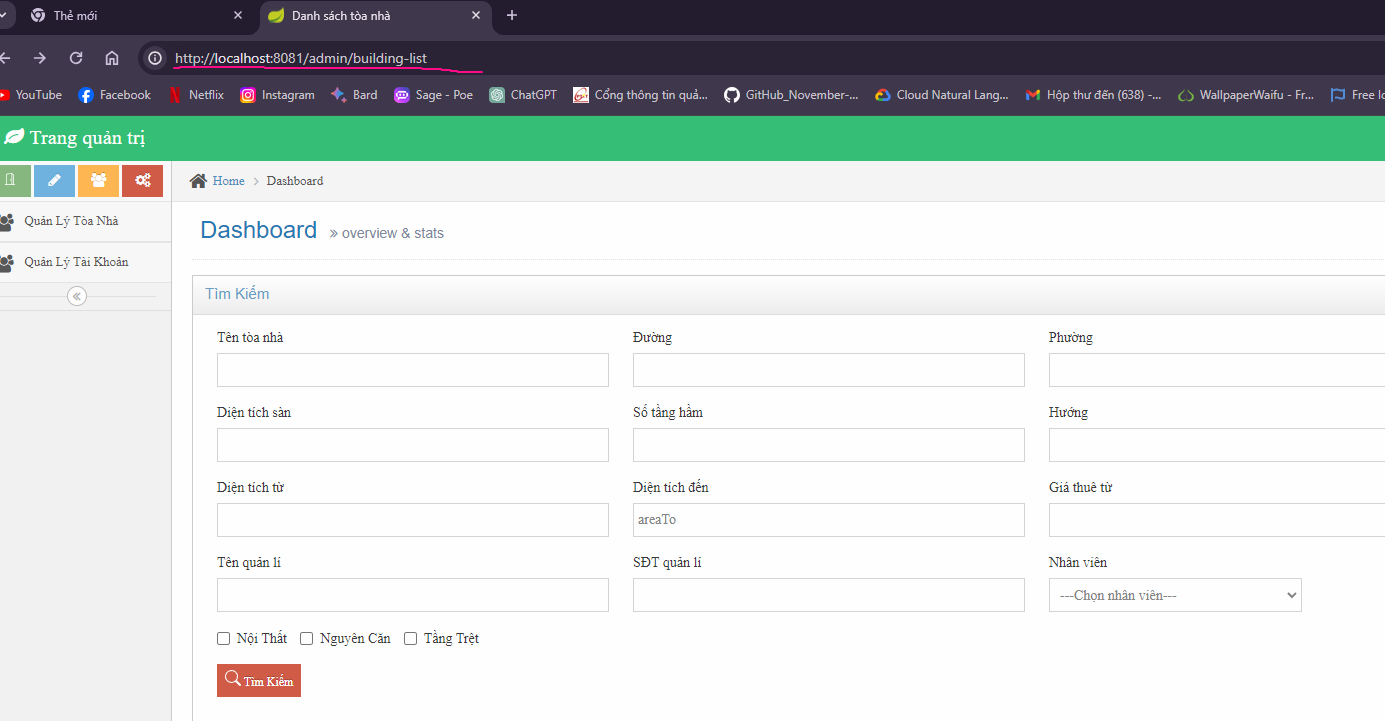
Khi sự kiện này được kích hoạt, form sẽ được gửi đi, tức là dữ liệu trong các trường input của form sẽ được đóng gói và gửi đến máy chủ để xử lý. Trong trường hợp này, khi người dùng nhấn nút có id là "btnSearchBuilding", sự kiện click được kích hoạt.

Sau đó, $('#listForm').submit() được gọi để gửi form đi bằng cách kích hoạt sự kiện gửi trên form có id là "listForm". Điều này có nghĩa là các giá trị trong các trường input của form sẽ được gửi đến máy chủ để xử lý.

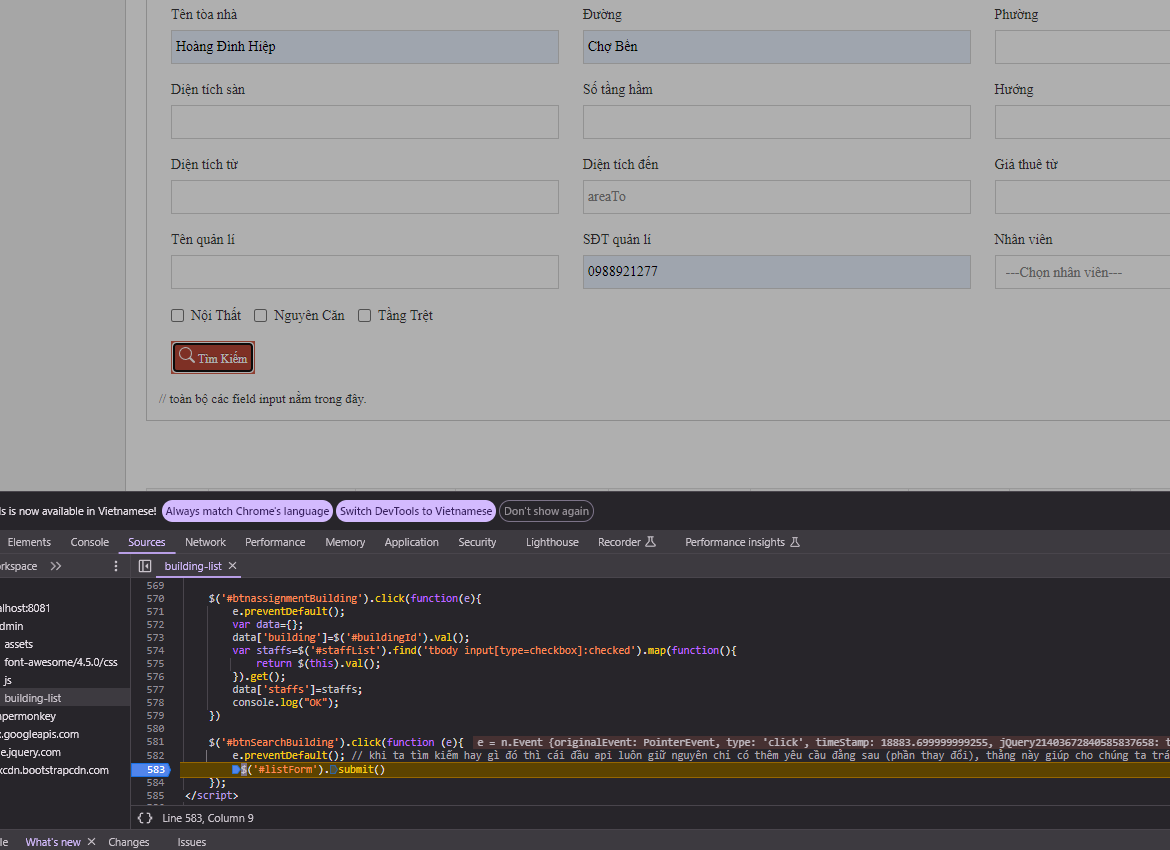


Thằng form:form này nó rất đặc biệt ở chỗ nó hỗ trợ chúng ta rất nhiều,hãy nhìn hai thuộc tính nó hỗ trợ action chính là cái api mà ta cần gửi đến và method chính là http method.

Lúc chưa gửi:



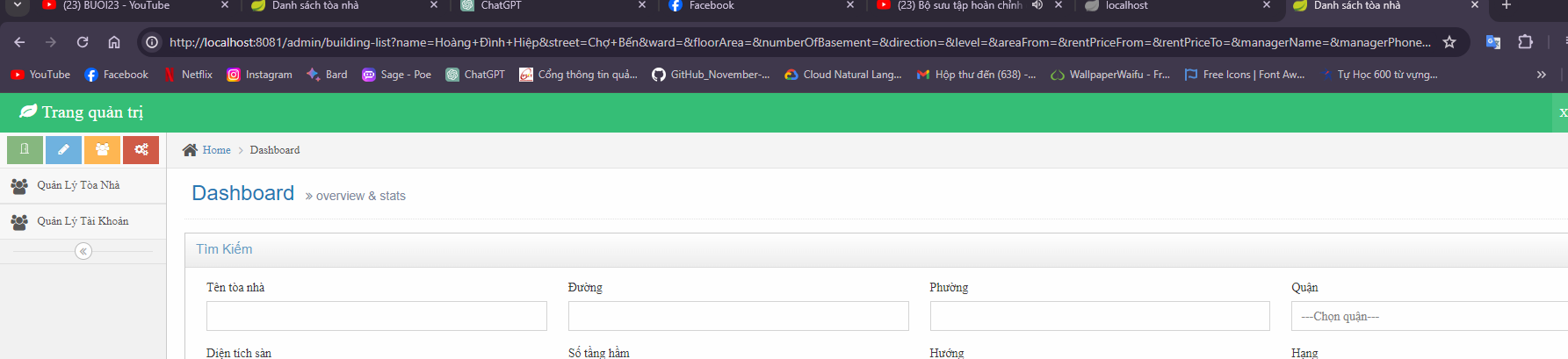
Bắt sự kiện click:



Bắt api nhận:

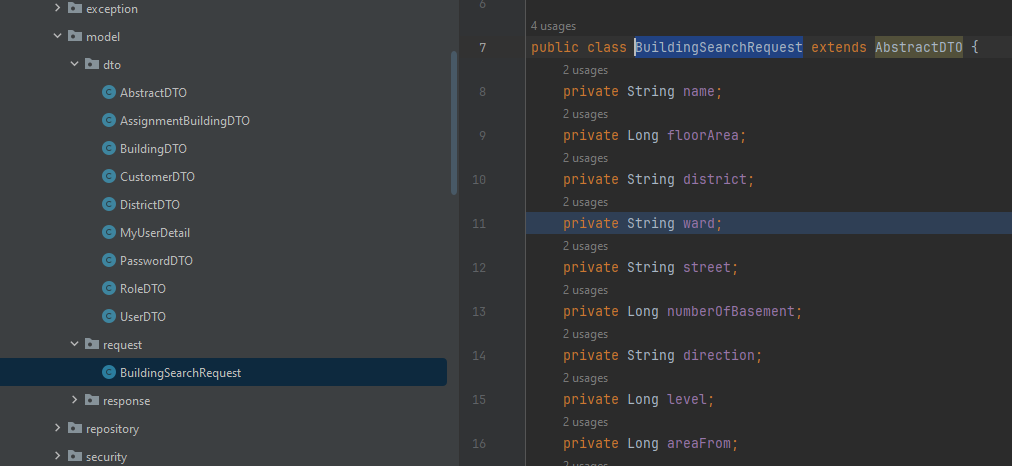


Hãy nhìn trên param



Phía trên đã test và đã đẩy đc lên param về controller nhưng giờ ta chỉ cần tạo một đối tượng để hứng data thôi:

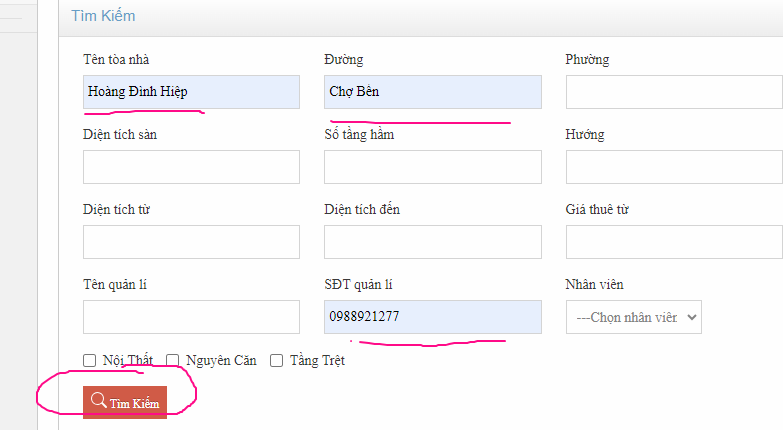
Tôi sẽ tạo một class như trc kia ta đã từng làm:



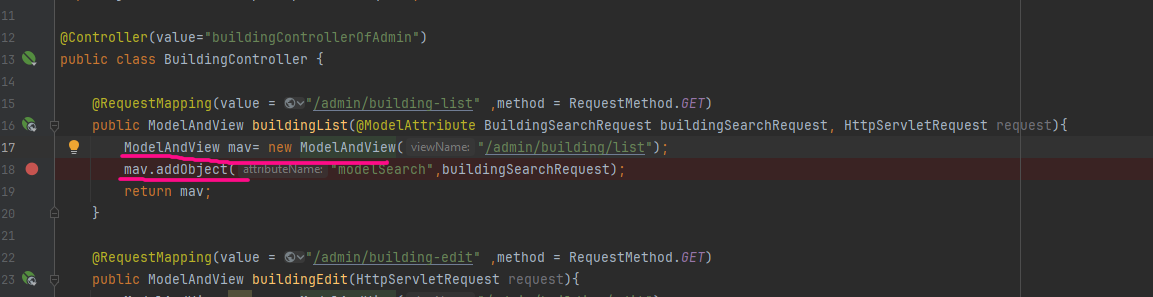
ở controller trc kia là nhận bằng map, giờ nhận bằng đối tượng thì ta có thêm một anotation mới là @ModelAttribute để dùng class hứng data



-Việc tiếp theo chúng ta cần làm là khi bấm nút để gửi data thì các điều kiện ban đầu trên các field sau khi tìm kiếm vẫn phải được giữ nguyên.



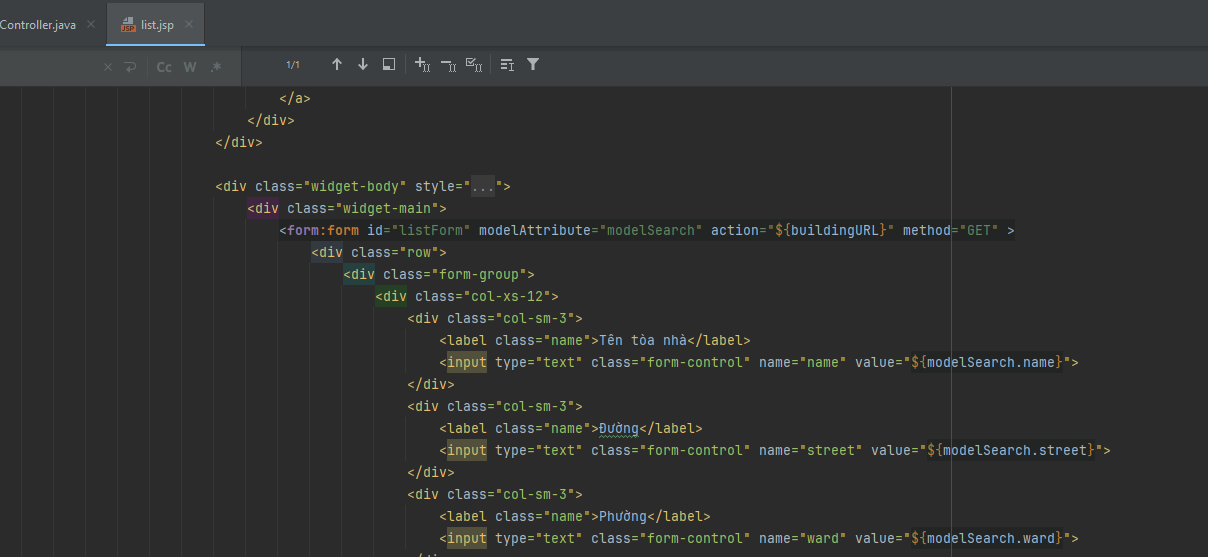
Để làm được điều này bạn hãy nhớ ta có ModelAndView của sevlet mà ta đã từng dùng để hiển thị 1 trang HTML , giờ ta có thể dùng phương thức .addObject của đối tượng thuộc ModelAndView để dán dữ liệu lại cho các filed:



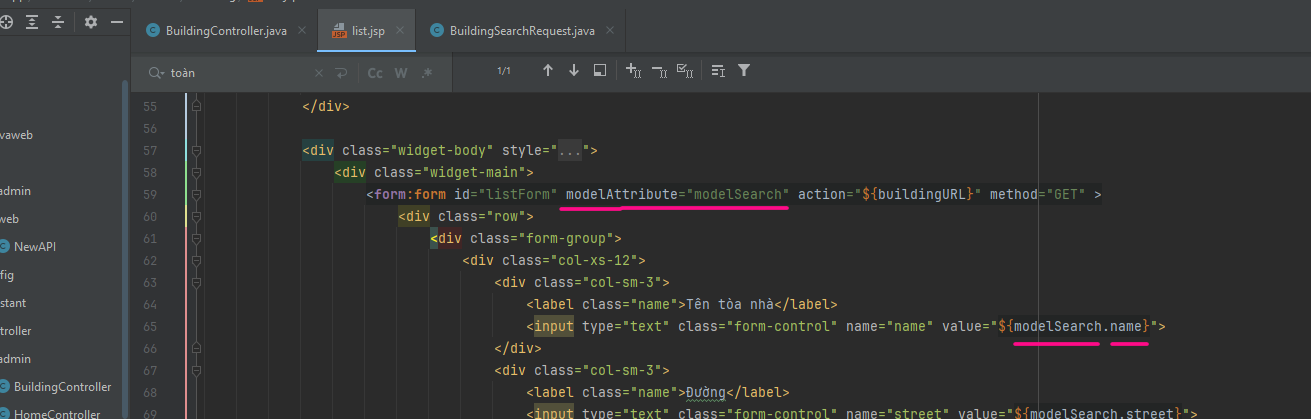
Giờ tôi sẽ giải thích giai đoạn đầu để làm được mong muốn trên:

mav.addObject("modelSearch", buildingSearchRequest); Đoạn mã này thêm một đối tượng có tên "modelSearch" vào ModelAndView, với giá trị là đối tượng buildingSearchRequest. Điều này cho phép dữ liệu các yêu cầu tìm kiếm được lưu trong buildingSearchRequest được chuyển từ phương thức này đến trang JSP để hiển thị cho người dùng.Có phải là ban đầu khi ta gửi data về từ client thì phía BE ta sẽ nhận bằng 1 đối tượng, giờ ta sẽ sử dụng chính đối tượng đó để hiển thị lại.

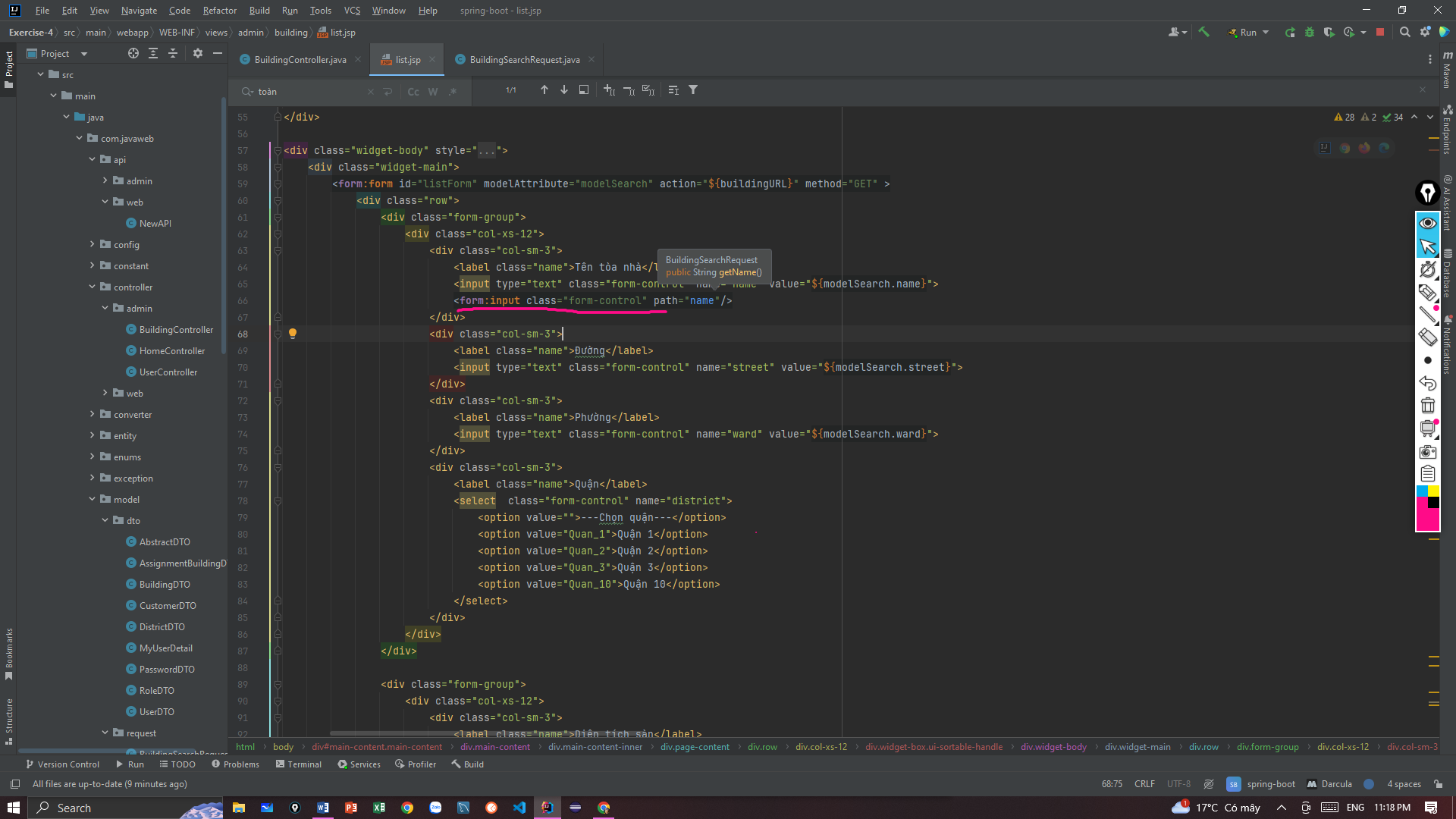
quay trở lại file jsp:



Giờ tôi sẽ nói lại tại sao dùng thẻ <form:form> mà không phải form bởi vì nó đc cấu hình từ spring framework nó hỗ trợ nhiều về java hãy nhìn thuộc tính modelAttribute=”name” khi khai báo thuộc tính này xong giờ ta chỉ cần gọi thuộc tính từ thằng đó để gán lại cho các value của thẻ input (hãy nhớ tại sao lại như vậy bởi khi ta nhập data vào field và gửi đi thì data đó chính là value trong thẻ input vậy đương nhiên muốn nó hiển thị lại thì ta chỉ cần gán lại value cho nó)

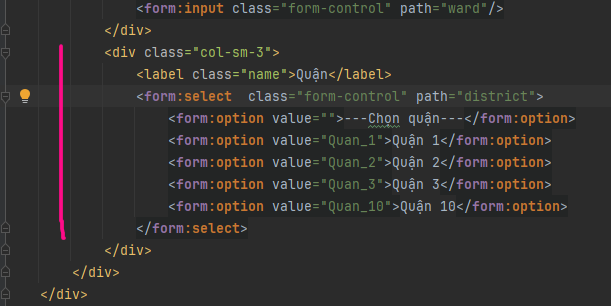


Nhưng giờ ta sẽ không dùng thẻ input như trên mà ta sẽ dùng thẻ <form:input />



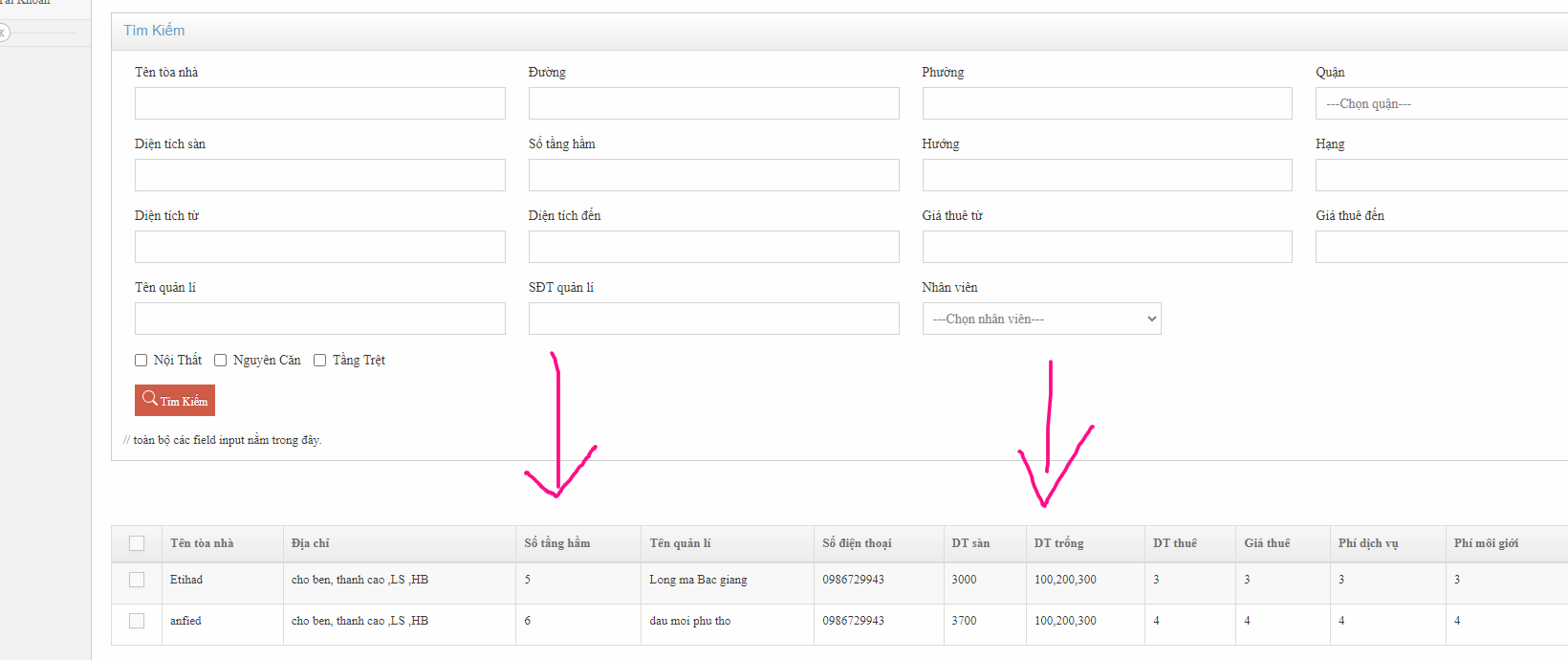
Path chính là cái key đó thằng này là theo phương thức get của đối tượng mà ta khai báo ở thẻ form:form

-sau đây ta cũng sẽ thay thẻ select bằng form:selectS



-Tiếp theo chúng ta sẽ sang đến phần Hiển thị danh sách các tòa nhà:

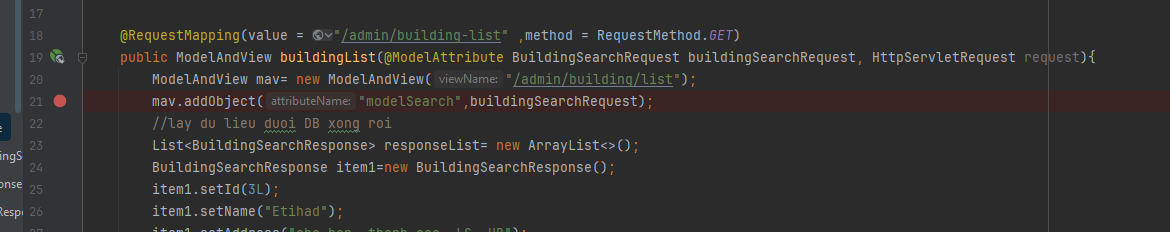
Thứ tôi muốn làm là nhập các field tìm kiếm theo yêu cầu và trả về những tòa nhà đó:



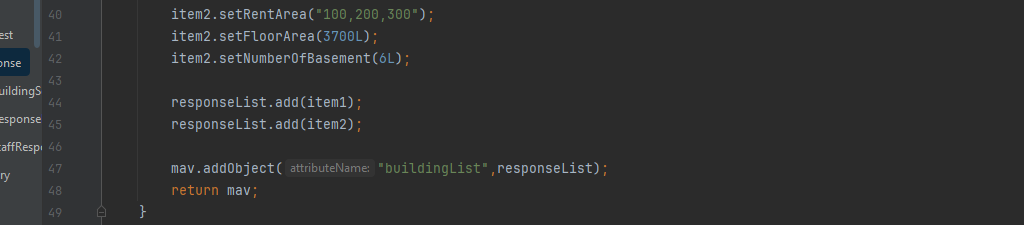
Về ý tưởng ta sẽ làm như sau ban đầu client gửi request và BE nhận bằng 1 model(object) công việc tiếp theo sẽ là cầm cục model đó lấy data từ DB thông qua mô hình MVC sau đó sẽ trả về 1 list các tòa nhà thỏa mãn sau đó ta chỉ cần dùng ModelAndView để sử dụng và gán nó sang các field.

Sau đây là ví dụ:

Vì đây là ví dụ nên tôi sẽ không lấy data mà tôi sẽ gán cứng , nếu theo quy trình là dòng số 22 sẽ trả về 1 list các data lấy từ DB.



Vì đây là ví dụ nên tôi sẽ không lấy data mà tôi sẽ gán cứng , nếu theo quy trình là dòng số 22 sẽ trả về 1 list các data lấy từ DB.

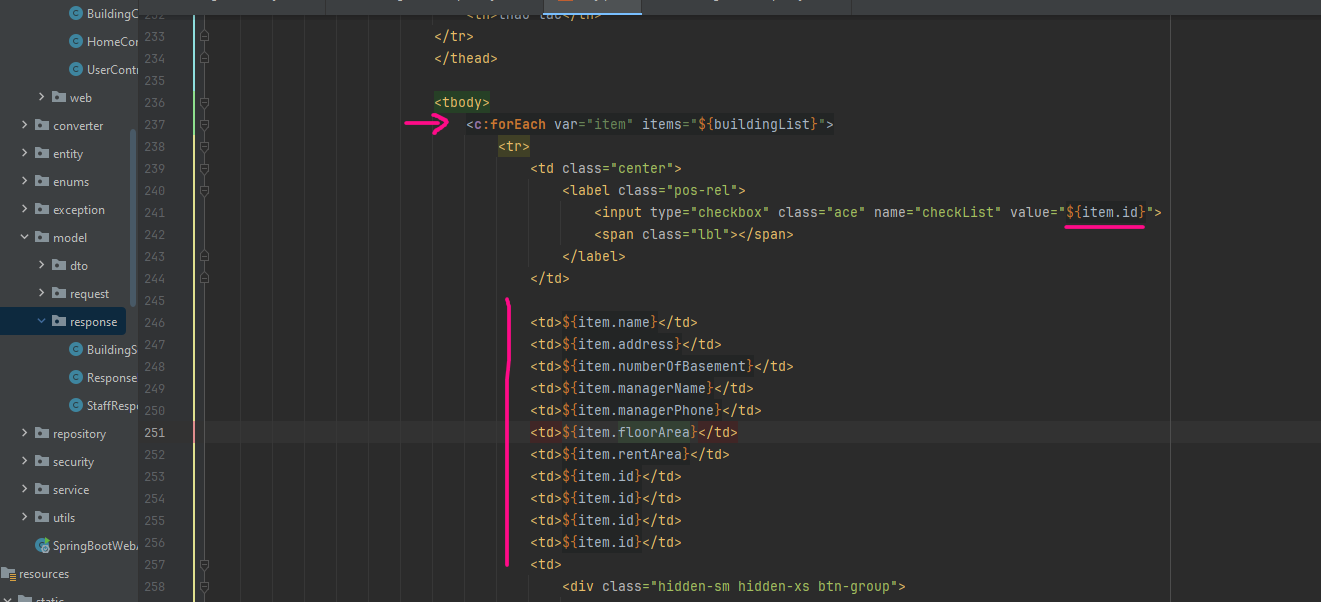


Sang đến file giao diện ta sẽ có 1 thẻ mới có tên là:

<c:forEach var=”item” items=”${ModelList}”>

Thằng này có thể hiểu nó là một cái vòng lặp duyệt qua từng phần tử trong model

Mỗi bước duyệt ta sẽ gán data cho các col

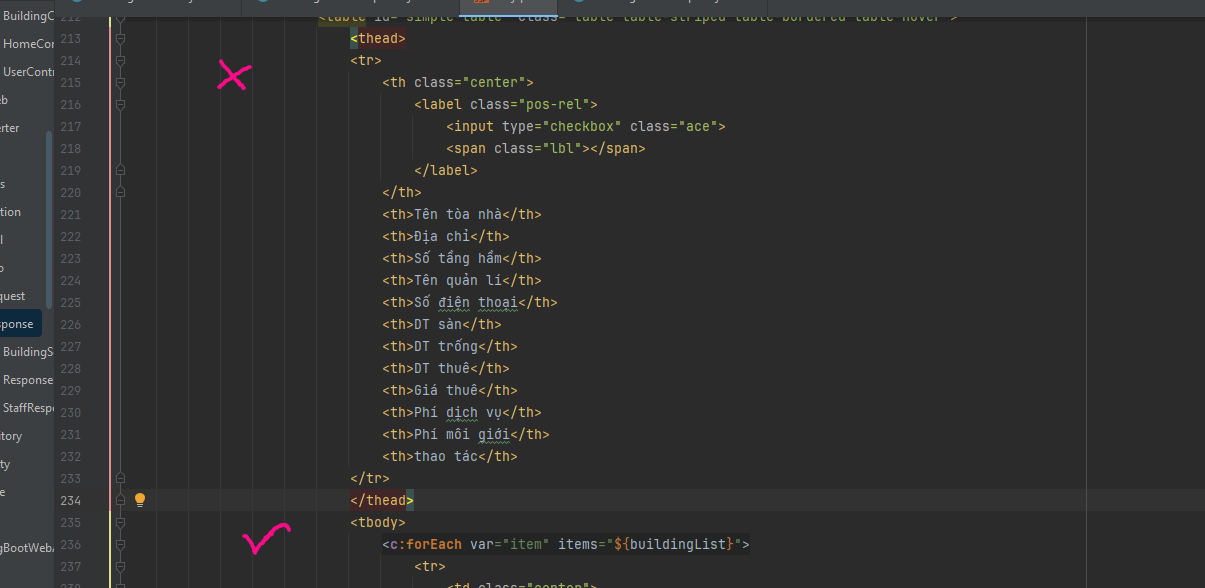


Hãy nhìn dòng 241 nó sẽ tương ứng với checkbox ở cột đầu tiên:

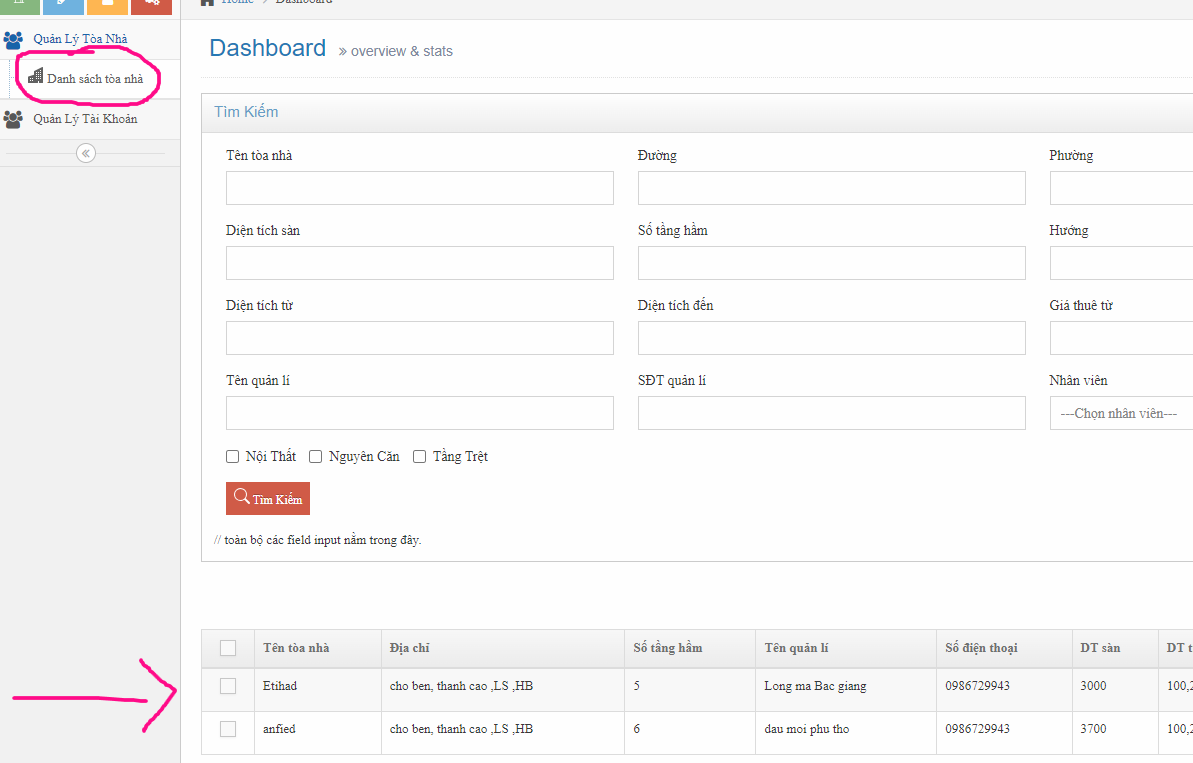


Mục đích set id cho nó là sau ta sẽ cần phải xóa hoặc sửa tòa nhà….

Và ta sẽ sử dụng c:forEach đối với phần tbody vì đây là phần thông tin còn ta không sử dụng nó bắt đầu từ thead nơi chứa thẻ tiêu đề th:

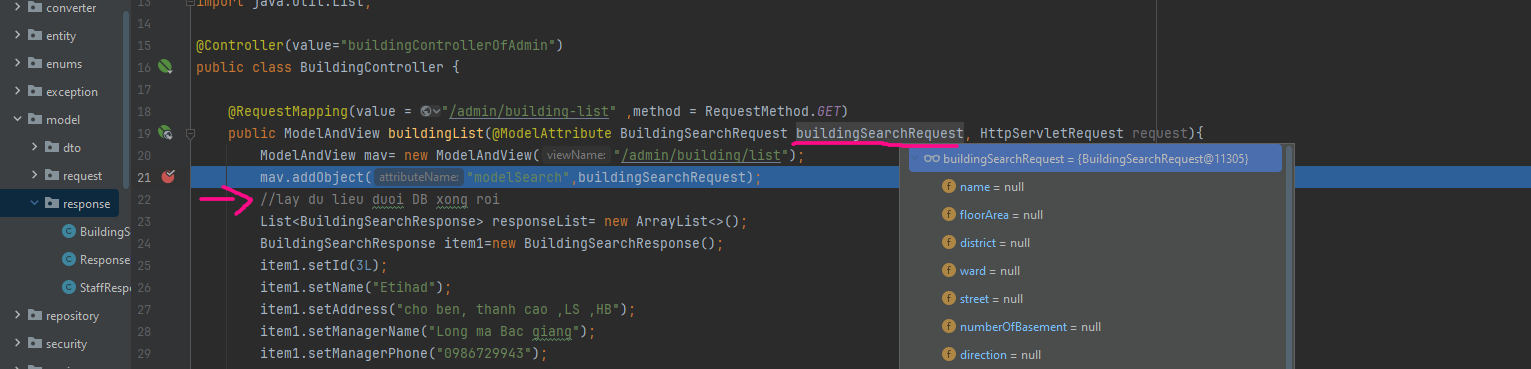


Một điều nữa đó chính là tại sao khi ta bấm vào “Danh sách tòa nhà” mà nó đã hiện ra toàn bộ tòa nhà mặc dù ta không tìm kiếm gì:



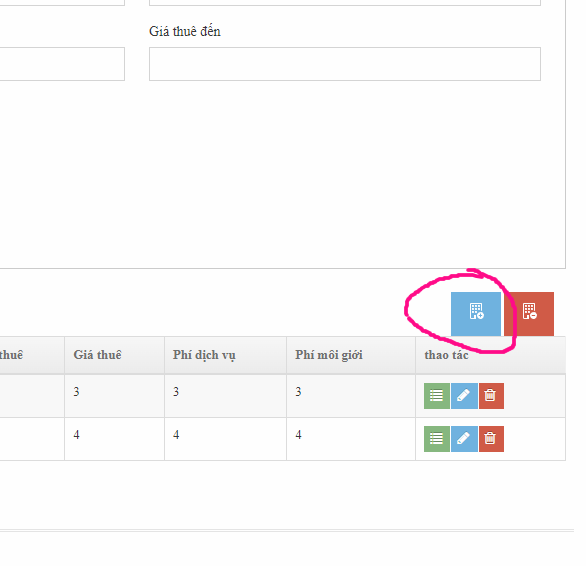
Tại vì như sau:

Sau khi login vào trang web và bấm vào “Danh sách tòa nhà” thì lúc này 1 yêu cầu đã được gửi đi về BE để hiện thị trang web lúc này data gửi về sẽ là Data rỗng vậy nên theo quy trình nó vẫn chạy theo mô hình 3 MVC (dòng 22) vì nó rỗng nên nó sẽ trả về toàn bộ các tòa nhà

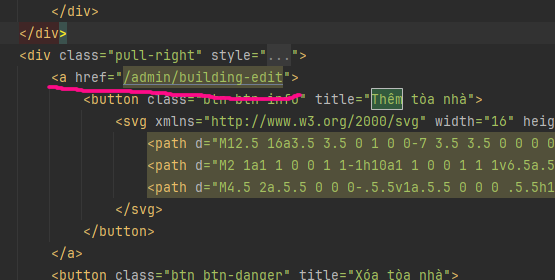


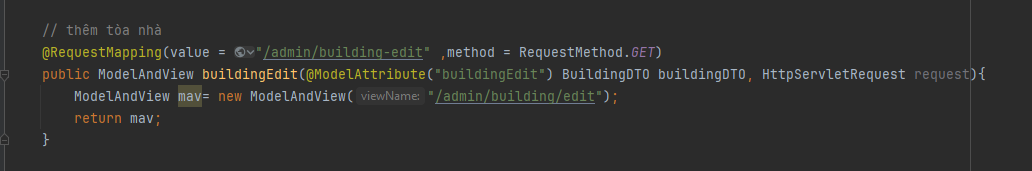
Còn sau khi điền đầy đủ vào các field và tìm kiếm thì lúc này data gửi về sẽ không rỗng và chạy theo mô hình 3 MVC để lấy data rồi hiển thị ra tương ứng thôi

-Sang phần thêm tòa nhà ( bước đầu) :



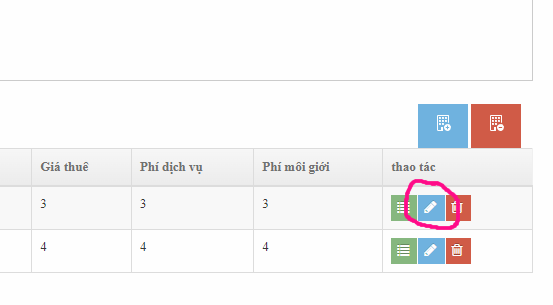
Ý tưởng khá dễ thôi: bắt sự kiện call api và xử lí.



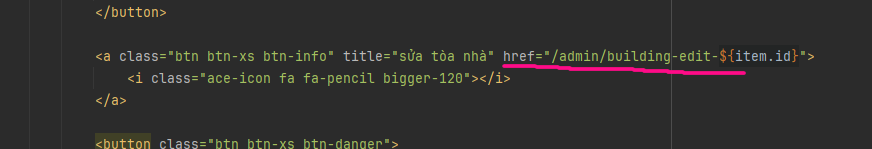


Nhưng trên mới là tạm thời thôi nha.

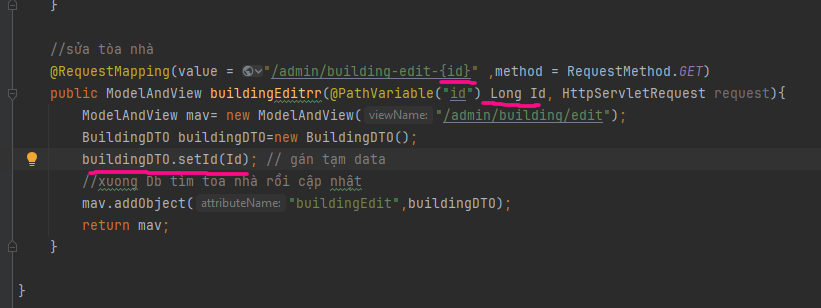
-Sửa tòa nhà:xử lí gần giống như thêm khác ở chỗ thêm thì không gửi id , sửa thì gửi id



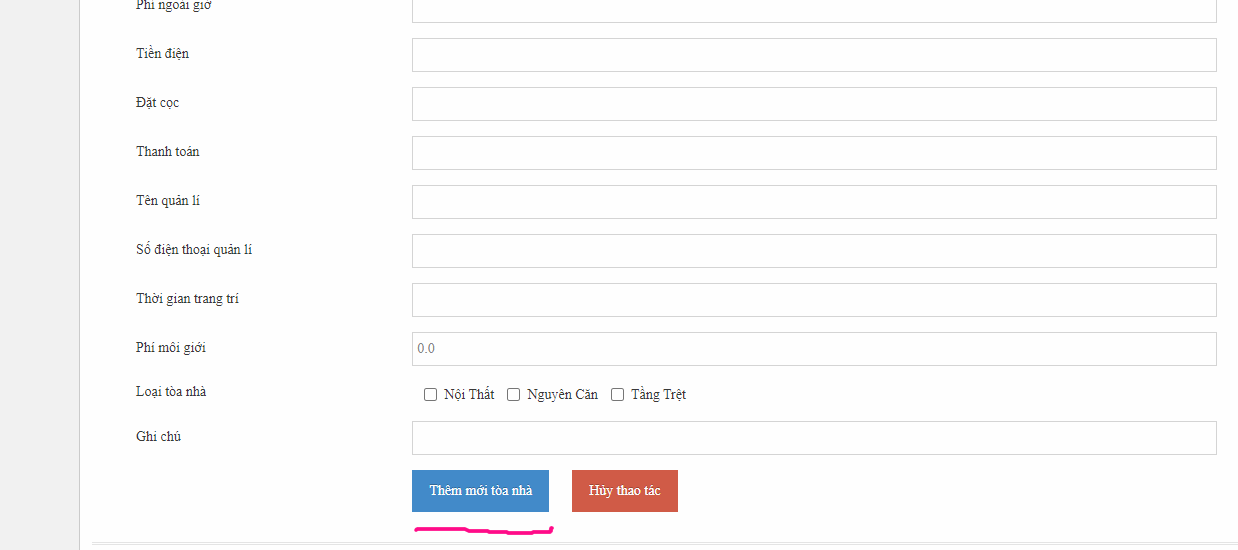
File menu list:



Controller:



Khi ta bấm thêm hoặc sửa tòa nhà thì thì sẽ đều hiện lên cái view sau:



Trên là dao diện khi ta bấm thêm tòa nhà ,vậy thì sửa tòa nhà cũng tương tự vậy khác là chữ “thêm mới tòa nhà” thành “cập nhật tòa nhà”

Làm cách sau: ta sẽ phân biệt nó bằng cách: khi thêm ta không gửi id và khi cập nhật ta sẽ gửi id, và bản chất khi ta call api thì nó đều gọi ra cái view chung bên trên vậy nên ta chỉ cần sử dụng <c:if>.

