Buổi 17

Spring Data JPA: cung cấp cho chúng ta những hàm tiện lợi để thao tác với cơ sở dữ liệu nhưng cũng sẽ có một vài hàm không có sẵn và chúng ta phải đi custom.

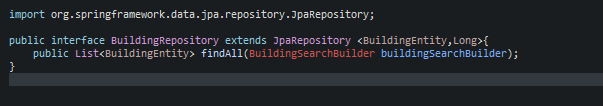
Tài liệu

<https://docs.spring.io/spring-data/jpa/docs/current-SNAPSHOT/reference/html/#jpa.repositories>

https://docs.spring.io/spring-data/jpa/reference/jpa/query-methods.html

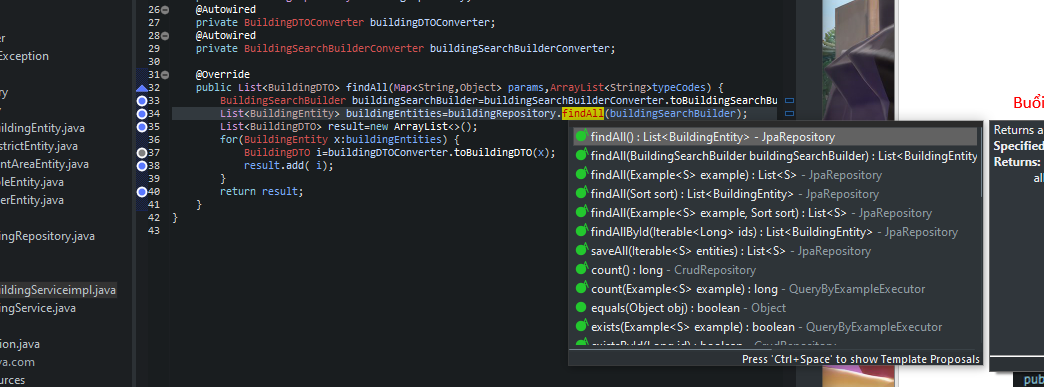
config:

ở tầng repository chỗ interface hãy extends JpaRepositoy<BuildingEntity,Long>



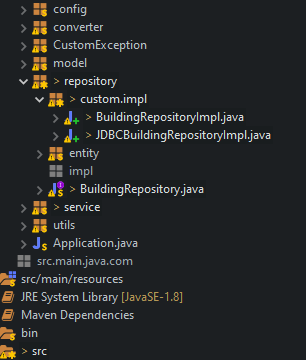
Tham số thứ nhất là Tên của class ClassEntity muốn tìm thứ hai là kiểu dữ liệu id của nó

Giờ hãy nhìn các hàm mà spring data jpa cung cấp:

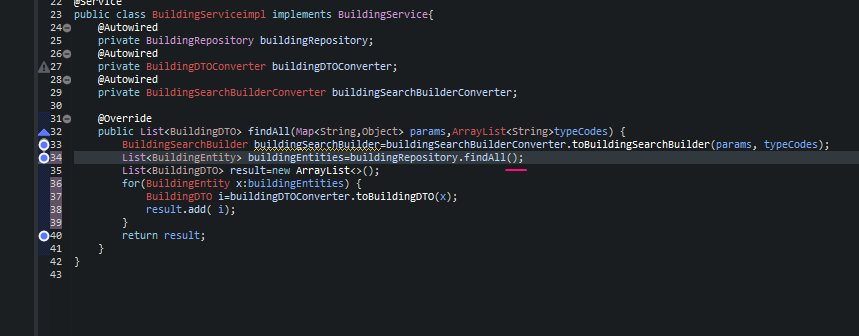


**Chúng ta sẽ cùng nhau demo test một vài hàm trước nhé**

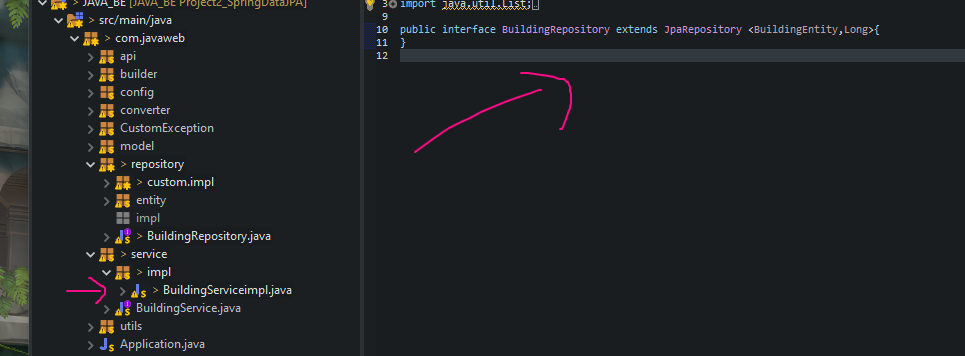
Đầu tiên ở tầng repo tôi không cần sử dụng đến các hàm mà trước kia ta implement từ interface để xử lý lấy data từ DB ,package impl trống không.



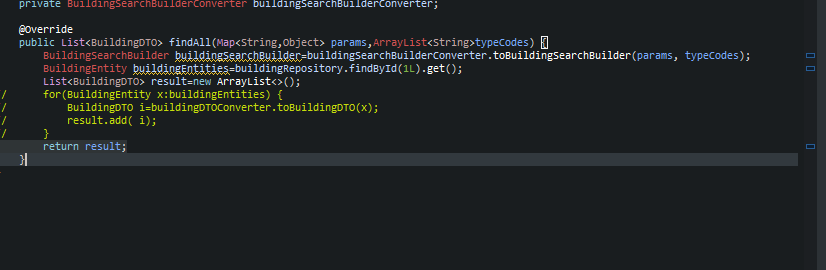
Giờ chúng ta sẽ quay trơ lại tầng sevice để tìm toàn bộ tòa nhà:



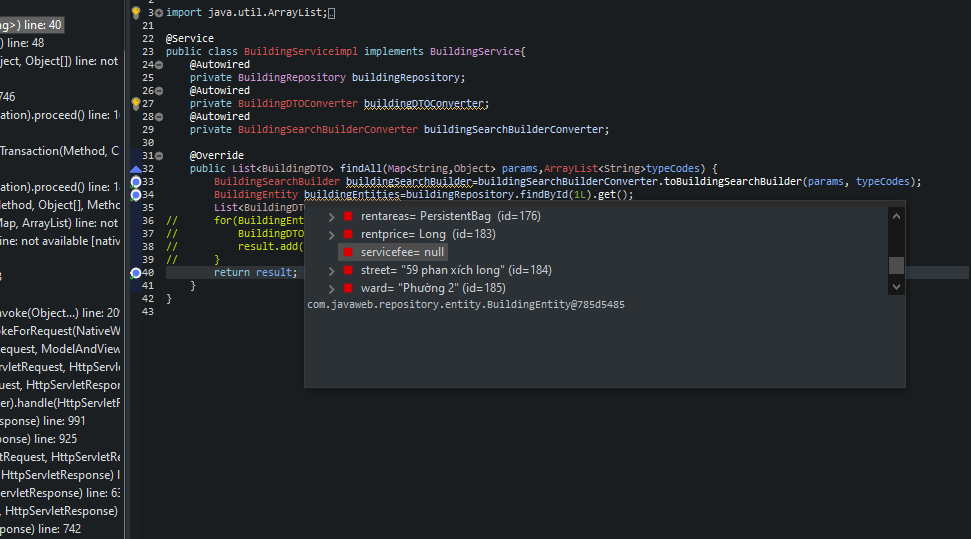
Giờ hãy so sánh đối số của hàm findAll hiện tại với hàm findAll ở trên cùng thì ta sẽ thấy thằng springdatajpa sau khi cấu hình xong chả phải truyền đối số gì đúng không thứ chúng ta cần làm chỉ cần cấu hình cho nó ở interface



Tiếp theo là hàm findById(id) tôi sẽ tìm kiếm tòa nhà với 1 id cụ thể với hàm này

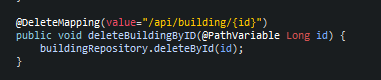


Bạn biết tại sao lại có .get() không thì tìm kiếm phải lấy ra chứ.

KQ: 

xóa tòa nhà bằng id:

đang ở tầng controller:

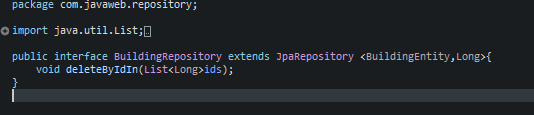


Nhớ autowired cái interface mà ta cấu hình



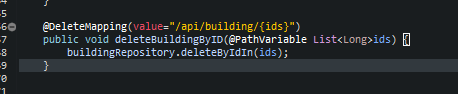
Giờ tôi sẽ đẩy nhiều id và nhận nó bằng 1 list rồi sau đó xóa cùng lúc như sau:

Đầu tiên chúng ta sẽ cần cấu hình :

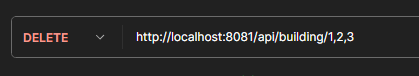


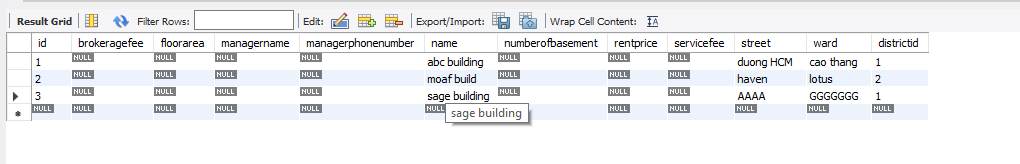
Hàm deleteByIdIn() này search ít tài liệu có , cấu hình cho nó ở interface

Quay lại với khâu xử lí ta chỉ cần gọi nó ra như sau:

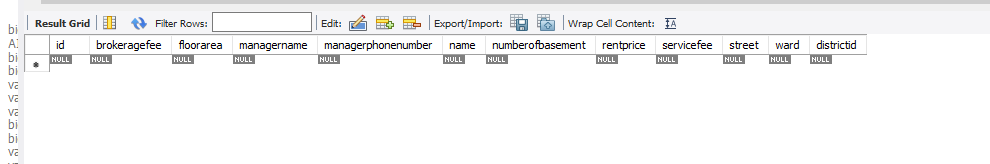


Giờ tôi sẽ truyền thử xóa 3 tòa nhà 1,2,3 cùng một lúc:

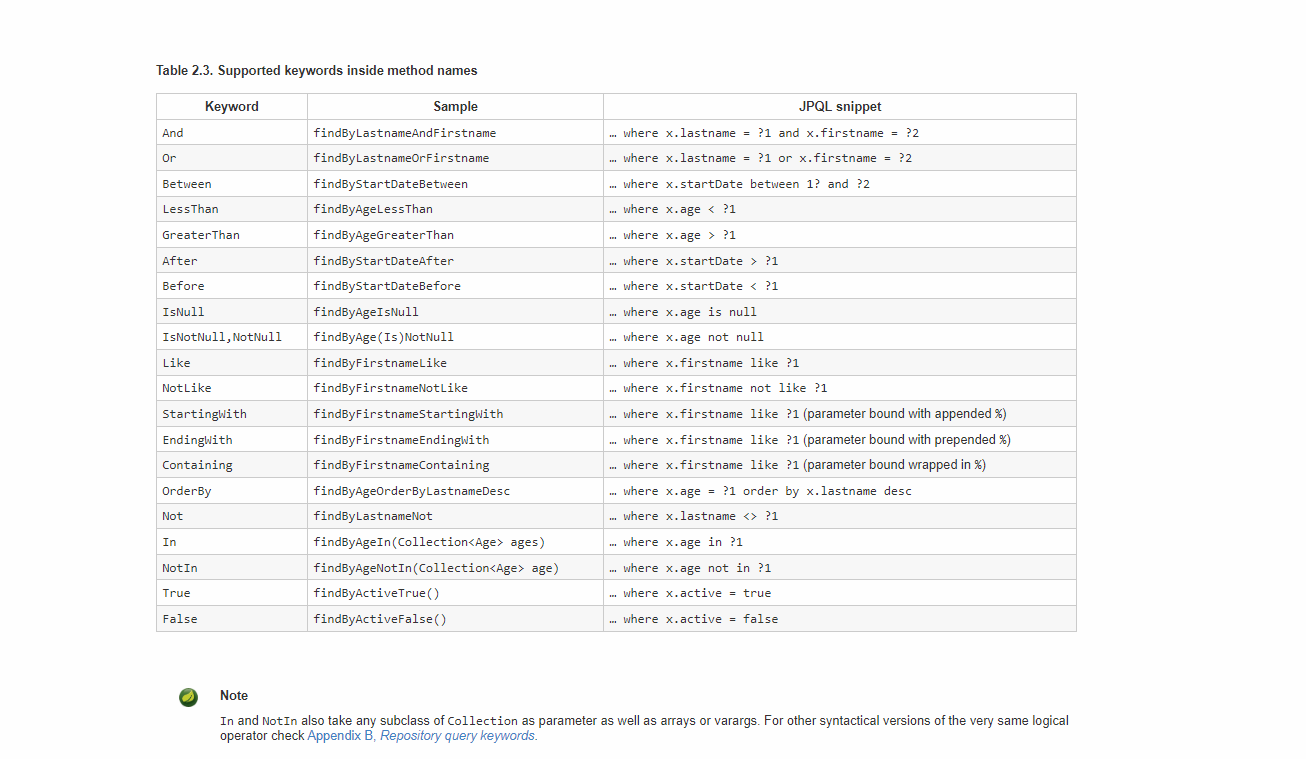




Sau khi chạy:

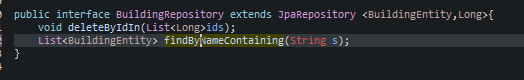


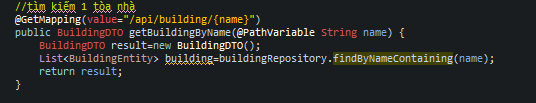
Trong bảng là một vài hàm mà springdataJPA đã xây dựng sẵn , chúng ta chỉ cần khai báo đúng cú pháp thì SpringdataJPA sẽ tự thêm querry cho chúng ta.



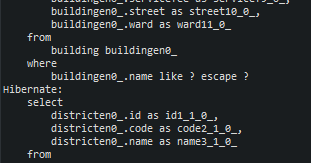
Demo thử hàm findByFirstNameContaining

Cấu hình :



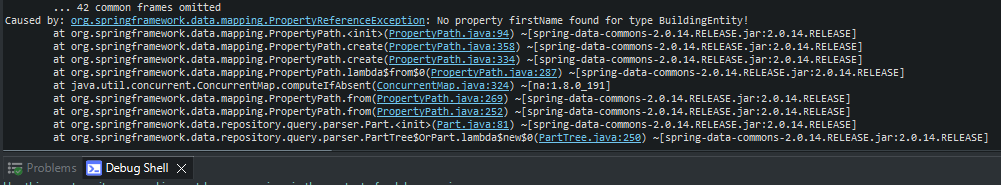


Code hibernate:

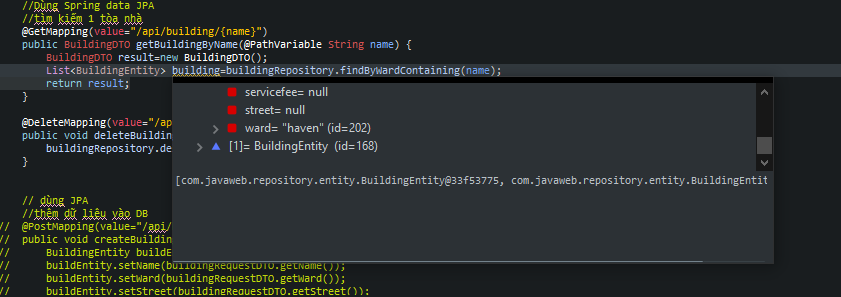


Nó tìm theo name nên nó dùng like để chọn ấy , nên sẽ trả về 1 list

Chắc b đang tự hỏi là tại sao bảo demo hàm findByFirstNameContaining nhưng nó lại thành findByNameContaining ? bởi vì trong bảng building của chúng ta để là name không phải first name nên khi đó nó sẽ không tìm đc, vì thế thôi đã thử đổi về thử thành Firstname thì nó xảy ra lỗi như sau:



Nó bảo là không tìm thấy thuộc tính first name vậy theo tôi nghĩ là thằng này tên nó đặt form sẵn rồi findByNameContaining kể cả ta dùng hàm này để tìm theo ward cũng được nhưng lúc này nó sẽ thành findByWardContaining ,tôi đã thử tìm theo ward và nó tìm được.

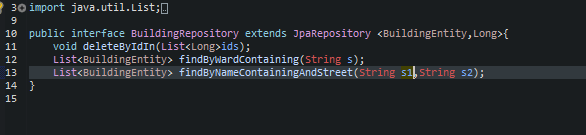


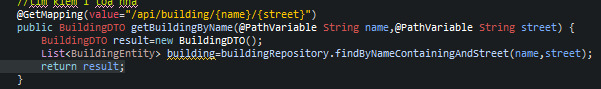
Theo tôi thì công thức của nó sẽ là findBy+Colname+Containing

Tiếp theo tôi sẽ tìm kiếm theo 2 field cùng một lúc.

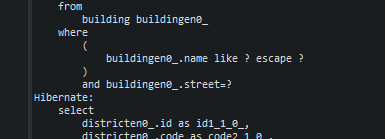
Nếu như ta tìm một field là findBy+Colname+Containing thì giờ chỉ cần thêm “AndColname2” nó chính là “And” đó

Cấu hình:

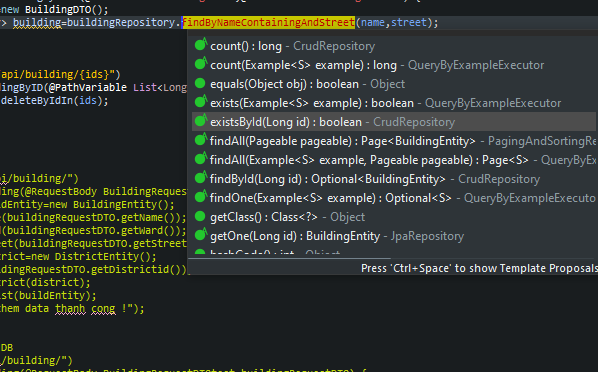




Code được sinh ra:

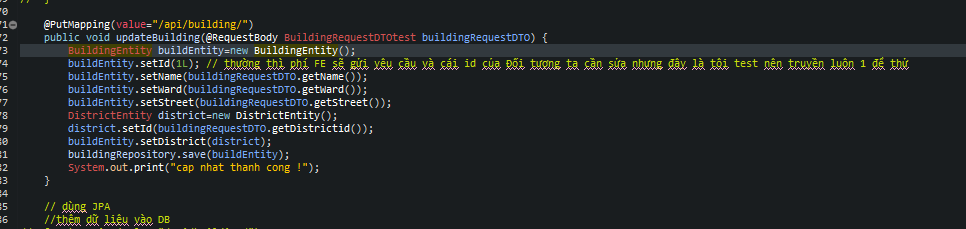


Ngoài ra còn một số hàm rất hay có thể tìm hiểu thêm như hàm existById nó sẽ check xem có bản ghi mang id đc được truyền có trong table hay không.



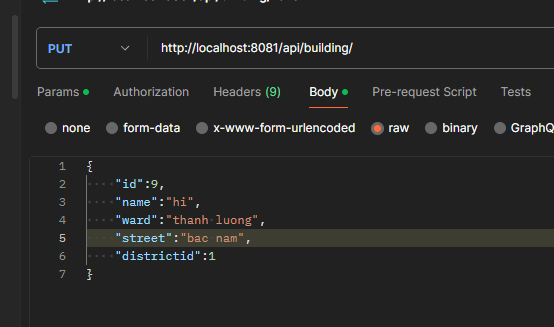
Tiếp theo là đẩy data vào bằng Save:

Nhận data bằng 1 đối tượng rồi lần lượt gán nó sang Entity



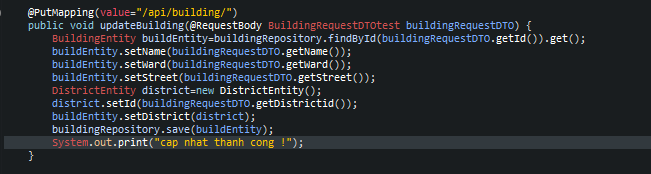
Cập nhật (update bản ghi):

Thường update sẽ gửi ở dạng body và kèm thêm 1 cái id của đối tượng mà ta muốn update:



Đầu tiên ta chỉ cần lấy tòa nhà mang id được gửi về rồi sau đó thực hiện update trên đối tượng đó và save thôi =))

(gần giống với insert data)



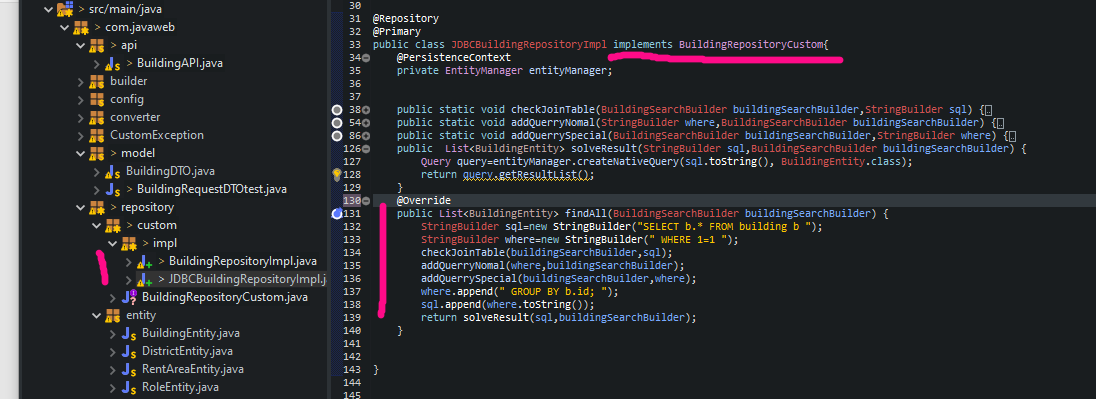
Tiếp theo chúng ta sẽ sang phần custom.

Tại sao phải có thằng này bởi khi lập trình và sử dụng Spring data JPA chỉ cần ta khai báo đúng cú pháp thì sẽ sử dụng được các hàm có sẵn của nó, nhưng sẽ có vài hàm không hỗ trợ,hay nói các khác là nó phức tạp, thường là chức năng tìm kiếm khiến chúng ta phải querry, thì ta phải đi custom nó (bằng JPA).

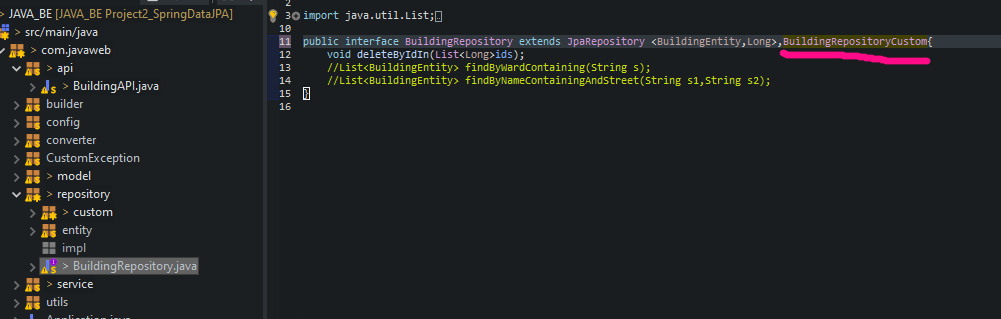
Đầu tiên ở tầng repo ta sẽ tạo một package tên là custom để chứa các impl :

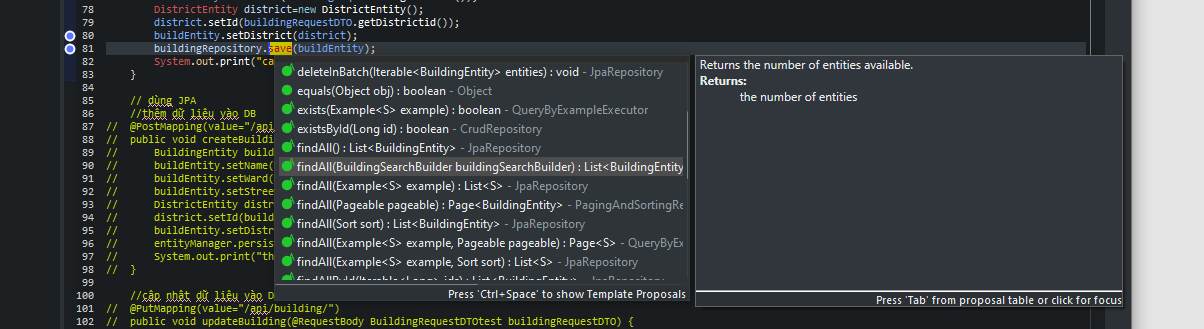


Trong đó chứa một interface có hậu tố là Custom và một pakage impl để chứa các cái class đc implement từ cái interface và các class đó cứ overide mà xây dựng các hàm custom.



Tiếp theo ta sử dụng đa kế thừa cho thằng interface mà ta cấu hình spring data JPA cho nó

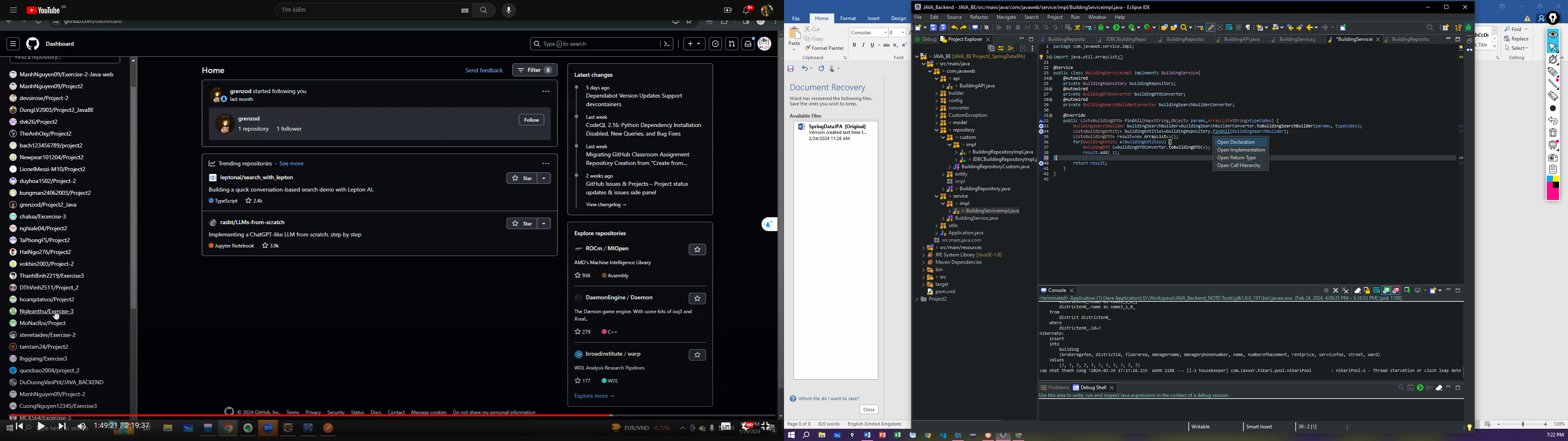


Giờ sang thử bên controller thử xem có chưa nhé 

Có rùi…

Đây chính là hàm findALL(với đối số là một BuildingSearchBuilder) tôi nghĩ sau này ở tâng sevice chỉ cần gọi hàm này rồi truyền đối số là BuildingSearchBuilder mọi thứ sẽ OK.

Sang tầng sevice



Nhìn ảnh trên bạn sẽ nghĩ cái findAll nó thuộc vào buildingRepository đúng không ?



Nhưng do thằng này nó extends từ thằng BuildingRepositoryCustom nên khi bấm open declaration nó sẽ ra thằng này:

