**BÀI 4: JPA VÀ DATA JPA TRONG SPRING MVC**

**Tái cấu trúc Project lại như JSP-Servlet**

Sau khi đã hoàn thành tái cấu trúc thì tiến hành thay đổi một chút bên file Controller. Tại đây tạo thêm một file NewsController để có thể điều khiển việc thêm xóa sửa bài viết

Ở phần trước, để nhận biết việc thêm xóa hay sửa dữ liệu thì ta cần phải có một constant để nhận biết và sử dụng if-else. (const LIST và EDIT)

A computer screen shot of text

Description automatically generated

Điều này rất bất tiện khi có nhiều controller hoặc nhiều thao tác khác nhau. Vậy nên trong Spring MVC có hỗ trợ việc trong một class này có thể có nhiều Controller khác nhau. Và mỗi Controller có thể điều khiển một thao tác riêng biệt

A black and white screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

**Sửa lại một chút về link để tối ưu việc truy cập**

Khi tiến hành truy cập vào một trang bất kỳ, sẽ thông qua url, và trong decorator phải khai báo url này. Điều này dẫn đến vấn đề là nếu có quá nhiều url trong Controller thì số lượng dòng code trong decorator.xml sẽ rất nhiều.

Do đó để tối ưu vấn đề này thì nên quy ước, ở đây ta quy ước mỗi url của admin (bao gồm tất cả các vấn đề thêm sửa xóa) đều bắt đầu với tiền tố là /admin. Còn web cho người dùng sẽ bắt đầu với tiền tố là /user

A screenshot of a computer program

Description automatically generated A black screen with text

Description automatically generated A black screen with text

Description automatically generated

File decorator sẽ có dạng thế này

A screen shot of a computer

Description automatically generated

**Ôn lại một chút về phần kiến thức đã học từ Servlet: Lấy dữ liệu từ request(url)**

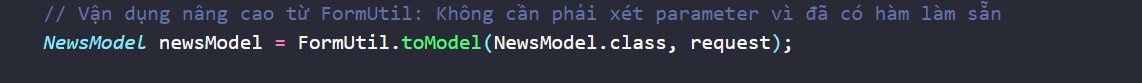
Tại NewsController, ví dụ ta có một url bất kỳ với các tham số như phân trang hoặc sắp xếp (các tham số này đều có thuộc tính trong AbstractModel) thì ta phải tự build một class gọi là FormUtil.

Class FormUtil này đóng vai trò caching các tham số trong url và đưa vào một biến tên model. Từ biến này thì có thể lấy ra các thông tin theo yêu cầu (quan trọng nhất là listResult). Từ đây có thể dùng model để tương tác trong view (thêm, hiển thị dữ liệu)

A computer screen shot of text

Description automatically generated

Đây là trong Controller, thay vì getParameter thì ta dùng cái này để tối ưu hóa code



**Cách lấy dữ liệu trong SpringMVC**

Khác với Servlet, SpringMVC có hỗ trợ một cơ chế tương tự như trên, đó là anotation @ModelAttribute. Anotation này đóng vai trò như một bộ phận caching, sẽ nhận các tham số từ url và cach vào model

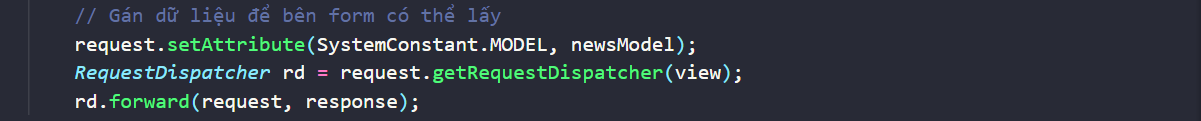
Như đã biết, ModelAndView sẽ trả về cho giao diện một đối tượng Model và một View để hiển thị. Khi đã có được view (đường dẫn đến file jsp đã được quy định bởi dispatcher) thì ta dùng phương thức addObject do ModelAttribute hỗ trợ (nó tương tự như hình ảnh ngay sau ảnh này – dùng trong Servlet).

*Trong SpringMVC*

A screen shot of a computer program

Description automatically generated

*Trong Servlet*



Ngay sau bước này thì ModelAndView sẽ có đầy đủ dữ liệu cho đến view để đưa ra màn hình trang web cho các thao tác bên trong sử dụng (dưới đây là ví dụ cho việc lấy các thuộc tính để phân trang)

A screen shot of a computer code

Description automatically generated