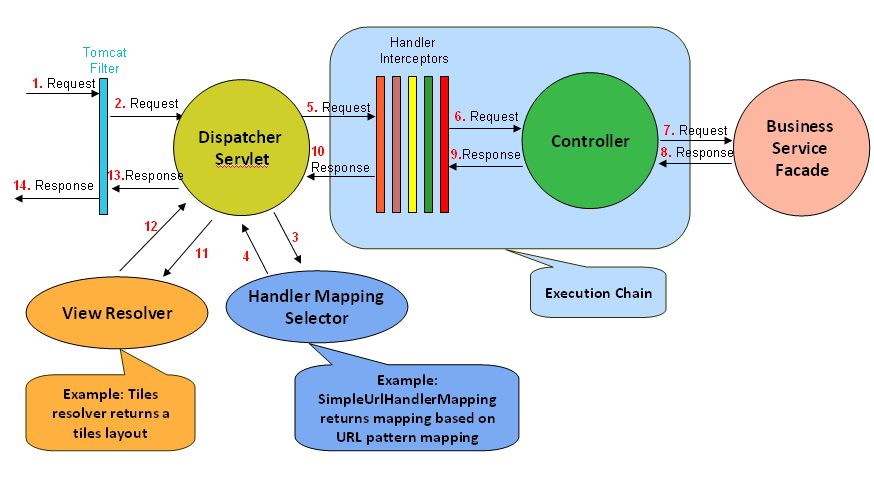
1. **Giới Thiệu :**

****

**B1 :** Yêu cầu chuyển tới Dispatcher Servlet. Tiến hành Handler Mapping kiểm tra với URL tương ứng thì action của controller nào được chạy.

**B2 :** Tiến hành gọi action của controller. Trước khi gọi controller, request phải đi qua **Handler Interceptors.**

**B3 :** Sau khi qua đến controller => Business Service Façade. Sau đó phản hồi trở lại controller.

**B4 :** Controller sẽ chạy 1 cái action – trả kết quả, làm gì đó … phản hồi về cho **Handler Interceptors.**

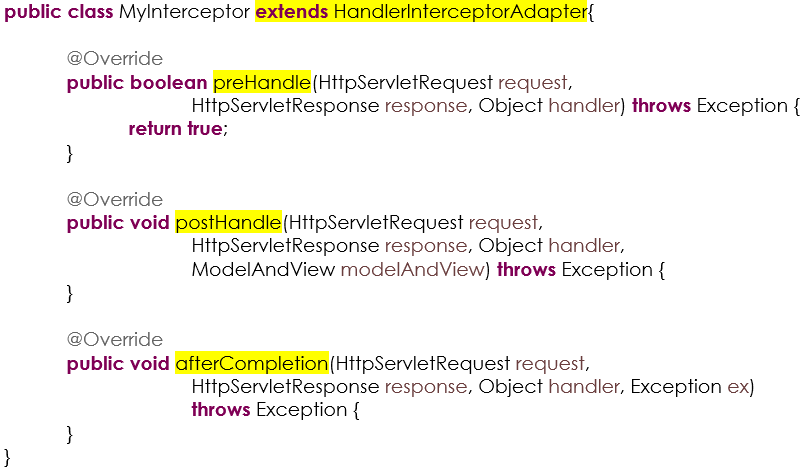
**B5 :** Sau đó phản hồi về **View Resolver.** Lúc này ta sẽ biết được file **JSP** nào sẽ chạy. Gọi view chạy. Sau đó trả về cho người dùng.

**Interceptors** khá giống filter. Chạy sau khi qua **DS.**

**Filter** chạy trước khi đến **servlet. Interceptors** chạy trước controller.

Sẽ chạy trước khi action chạy và sau khi action chạy.

1. **Cấu trúc Interceptor :**

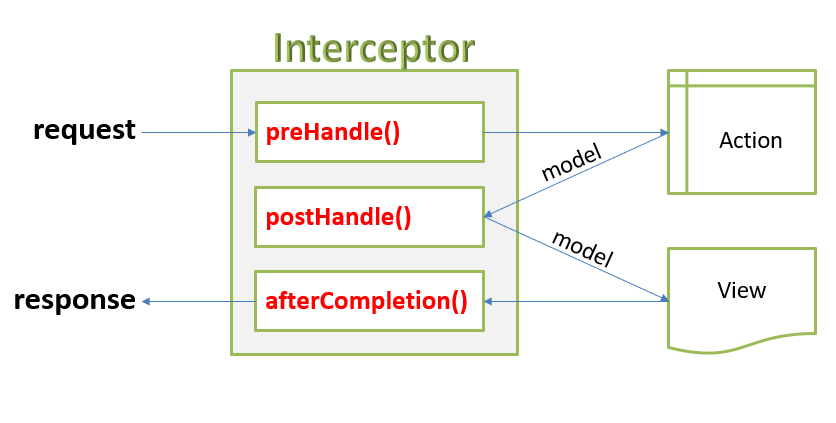


\_ Kế thừa HandlerInterceptorAdapter và override các method bắt buộc.

\_ **preHandle :** chạy trước khi tới action.

\_ **postHandle** : chạy sau khi ra khởi action.

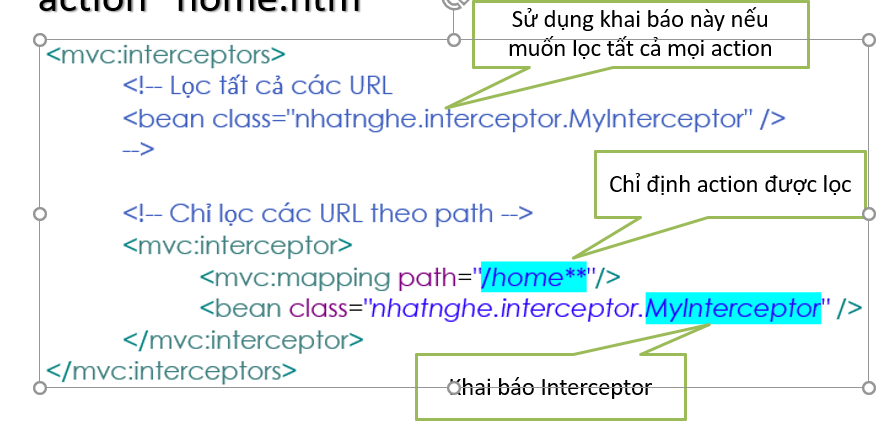
\_ **afterCompletion :** chạy sau khi view thực hiện xong.



**Ứng dụng :**

\_ Truyền dữ liệu từ controller => view ( tất cả view ) : viết ở **pre, post.**

1. **Cấu hình : 2 cách viết**

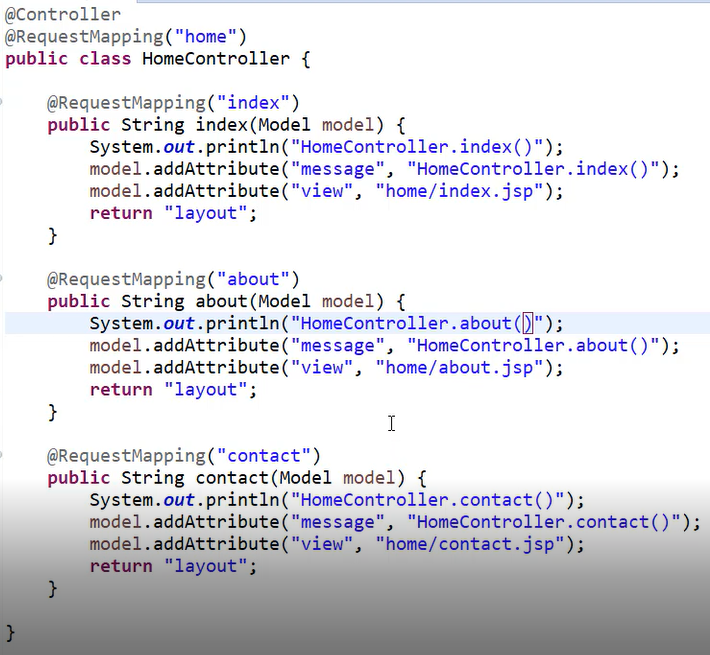


**\_ TH1 :** lọc tất cả mọi action

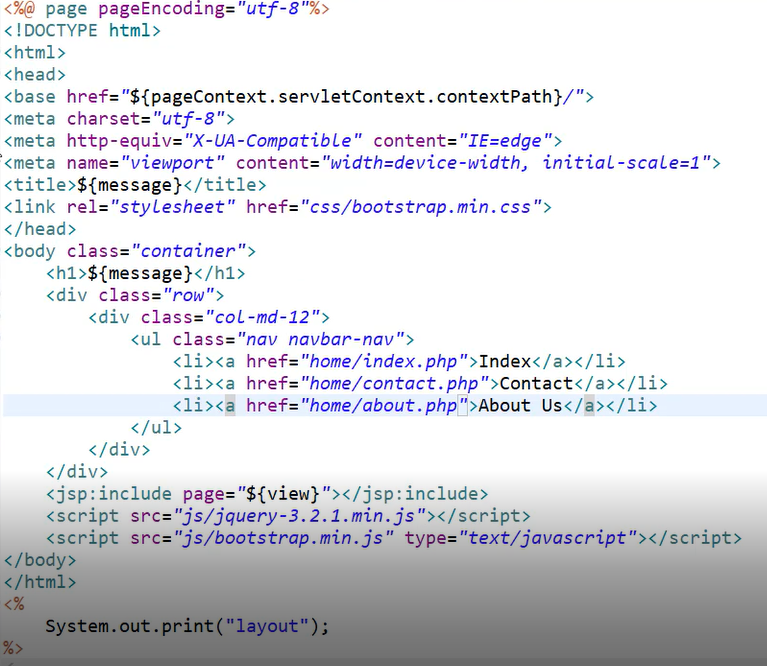
**\_ TH2 :** chỉ định action sẽ lọc.2 dấu \*\* có nghĩa là không giới hạn dấu **/**, bao nhiêu cấp cũng được. 1 dấu \* là đại diện 1 nhóm ký tự.

1. **Thực Hành :**

\_ Tạo controller :

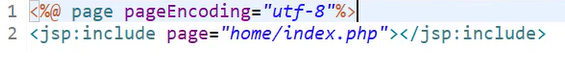


Tạo layout

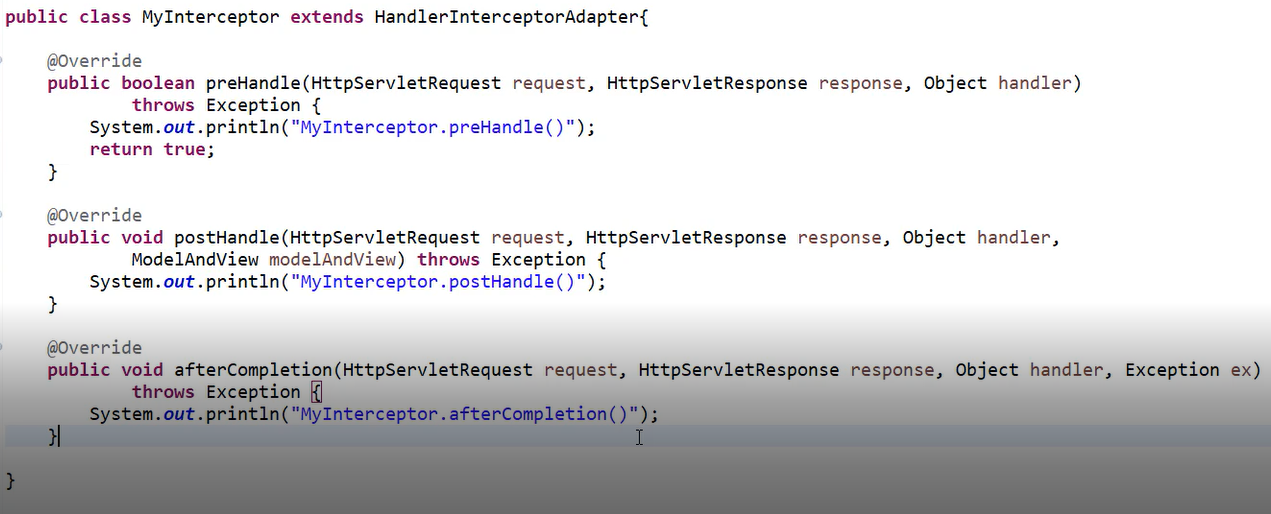


Tạo các trang tương ứng : index, contact, about.

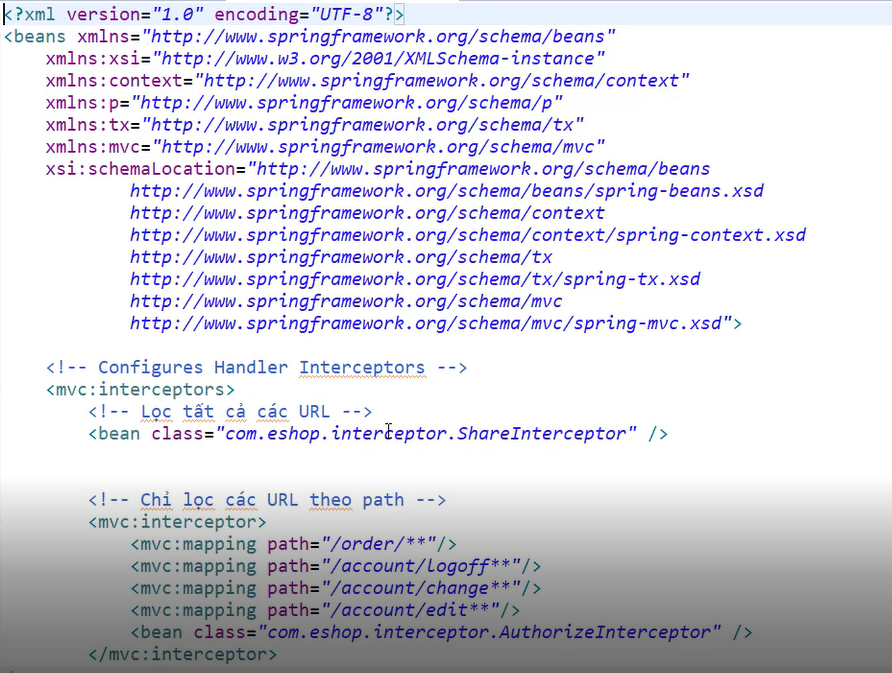
Tạo trang index ngoài cùng để khi chạy project lên nó include action – controller khác: **index.jsp** nằm ở trong WebContent



\_ Tạo interceptor : MyInterceptor



\_ Cấu hình interceptor : **spring-config-interceptor**



Ta có thể chỉnh sửa lại cho phù hợp.

Chạy & Kiểm tra

1. **Thực Tế :**

\_ Đăng nhập mới được vào một số trang :

Register

Login

**Forgot-password**

**Logout**

**ChangePassword**

**EditProfile**

**Checkout**

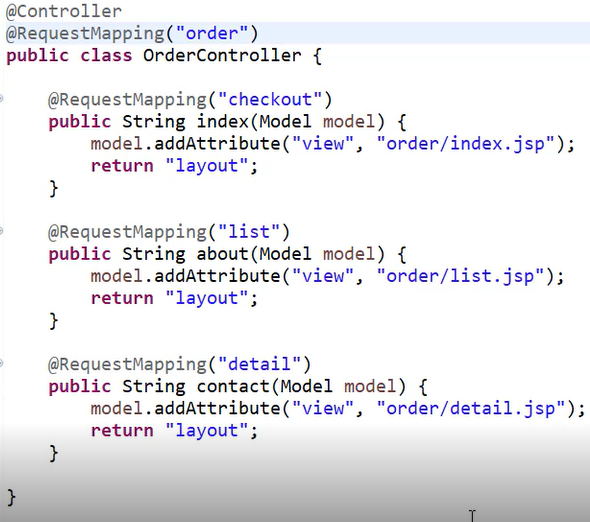
**MyOrderList**

**MyOrderDetail**

**MyProductList**

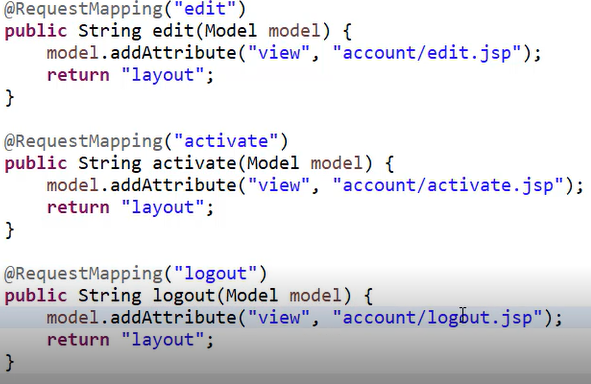
\_ Chưa đăng nhập cố tình gõ url vào thì chuyển trang trang login.

**B1 :** Tạo OrderController



**B2 :** Tạo AccountController





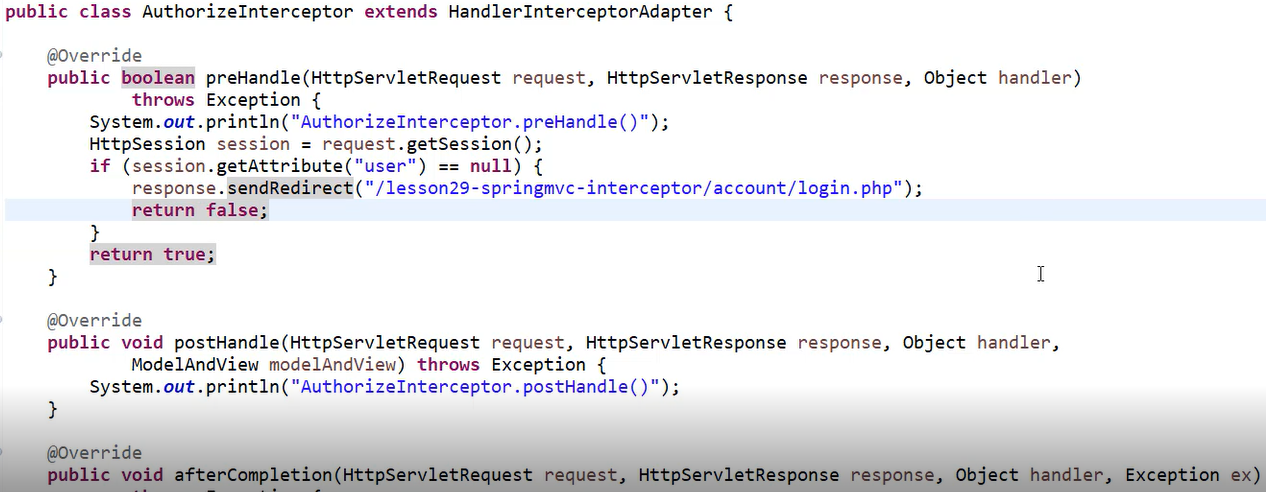
Thêm các liên kết tại layout : các trang order và account.

**B3 :** Tạo view cho các trang account**,** order

**B4 :** Tạo interceptor : **AuthorizeInterceptor**

Ý tưởng : kiểm tra xem trong session có attribute là user hay chưa ?

Nếu chưa có chuyển sang trang login.



\_ Xử lý tại preHandle. Return false là không cho đi tiếp.



Đây là contextPath. Ta nên xử lý bằng code java để tránh sai sót bằng đường dẫn.

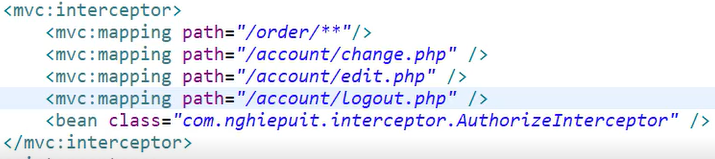
Như base href ở ngoài.



Lúc này đây là đường dẫn động. Không quan tâm tên dự án là gì nữa.

**B5 :** Khai báo cấu hình







**B6 :** Chức năng login :

\_ Cấu hình hibernate

\_ Tạo entity

\_ Tạo form login

\_ Tiêm @**Transactional** và @Autowired **SessionFactory** để làm việc với DB.

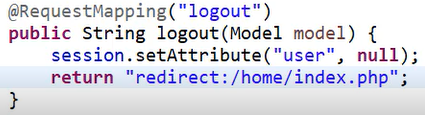
\_ Tiêm @Autowired @**Session** để làm việc với session.

\_ Tạo 2 action method @GetMapping login và @PostMapping



**B6 :** Chức năng logout :

\_ Hủy session quay lại trang chủ



**B6 :** Hiển thị menu theo trạng thái đăng nhập

**B7 :** Sau khi đăng nhập quay lại trang trước đó :

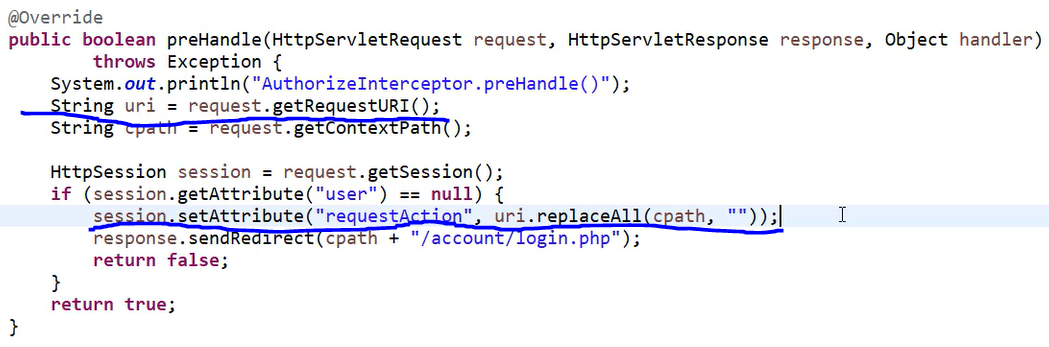
**\_** Ý tưởng : Trong interceptor trước khi chuyển sang trang **login** ta phải lưu trang đó lại trước. Sau khi đăng nhập xong. Kiểm tra đường dẫn có lưu hay không ? Nếu có chuyển trang.

**URL :** Toàn bộ đường dẫn.

**URI :** Là từ . Log ra xem.

**\_** Thực hiện : Lấy URI. Xóa ContextPath. Ta được action. Lưu vào session

Tại Interceptor :



Sau khi login thành công gọi session này ra

