**BÀI 5.3: SPRING SECURITY**

**Cấu hình Security**

Đầu tiên về cơ chế thì Spring Security khá giống Sitemesh, đó là thực hiện filter theo url. SS sẽ kiểm tra và duyệt các url mà ng dùng request vào, sau đó xem url nào có tiền tố đủ điều kiện để đăng nhập vào một vai trò nhất định nào đó

Do đó phải khai bao một đoạn filter trong file web.xml và contexparam

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

Tạo thêm file security.xml và thực hiện cấu hình file này: Phải thực hiện lần lượt Authentication và Authorization

Sơ lược lại một chút về cấu hình ở khóa trước. Phần đăng nhập sẽ được định nghĩa trong doPost của Controller. Sau đó tiến hành kiểm tra action, lấy ra model đăng nhập vào thông qua hàm toModel của FormUtil. Sau khi đã có đối tượng đăng nhập là model thì tiến hành Authentication, và khi đã xác định tài khoản đó tồn tại thì sẽ Authorization

A computer screen shot of a program code

Description automatically generated

Ở trong form login sẽ submit một cái form để gửi lên server thông tin của tài khoản đăng nhập. Và ở phần đăng nhập tài khoản, action và 2 input thể hiện 2 giá trị login của form sẽ mang thông tin của miền j\_ bên file security.xml



A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

+ login-page truyền vào là link url ở controller gọi ra trang login

+ Nếu đăng nhập thành công thì sẽ gọi ra đoạn authentication-success-handler-url, sẽ được xử lý sao

+ Nếu lỗi thì sẽ gọi ra authentication-failure-url với biến incorrectAccount

+ Tự động thoát trong một thời gina nhất định không dùng gì đến web thì sẽ tự động logout bằng biến invalid-session-url

+ Tiếp theo là cấu hình để có thể hiểu MD5

+ Sau đó gán các đường dẫn các hàm

Tiếp theo tạo trong impl của service class CustomUserDetailsService với nhiệm vụ là nó sẽ add tất cả thông tin của user vừa Authentication thành công vào một session để một tí còn có thể dùng lại. Nhưng trước đó tạo một UserRepository (có thể custom lại thông qua class MyUser)

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

Tiếp tục tạo pakage service và trong đó khởi tạo class CustomSuccessHandler để xử lý Authorization khi đã Authentication thành công. Có nghĩa là khi Authentication thành công từ class CustomUserDetailsService và thông tin đã được thêm vào session rồi thì class CustomSuccessHandler sẽ sử dụng session đó để tiến hành phân chia Authorizaion theo từng role

+ Tạo mới một pakage Util và tạo class SecurityUtil có chức năng lấy ra list roles từ UserEntity (ở quan hệ ManytoMany)

A computer screen shot of a code

Description automatically generated

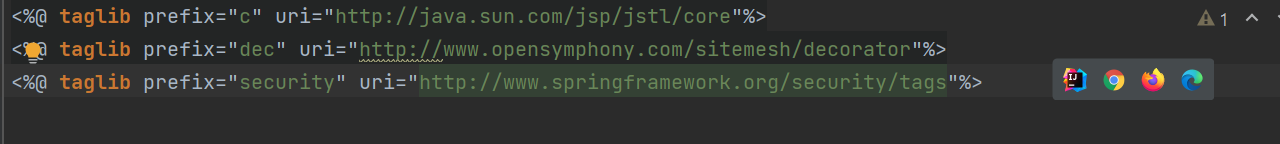
+ Định dạng Alert khi đăng nhập sai hoặc cố gắng truy cập vào một link mà không cho phép đăng nhập trong form

A screen shot of a computer program

Description automatically generated

**Xử lý sau khi đăng nhập thành công**

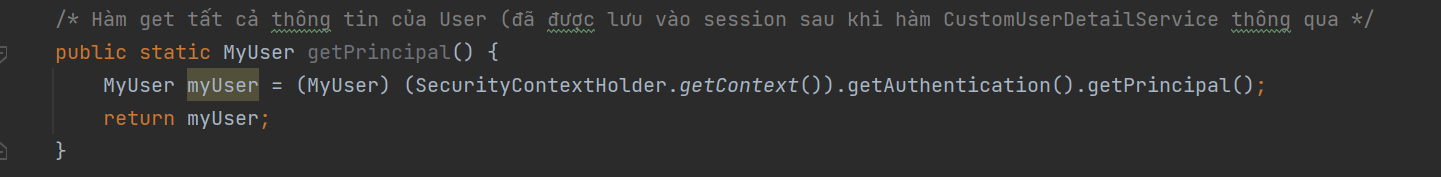
Đầu tiên cần phải lấy được thông tin của user vừa đăng nhập để in ra màn hình thì cần sử dụng cú pháp Spring Security trong jsp. Do đó phải add thư viện maven và khai báo trong taglib.jsp



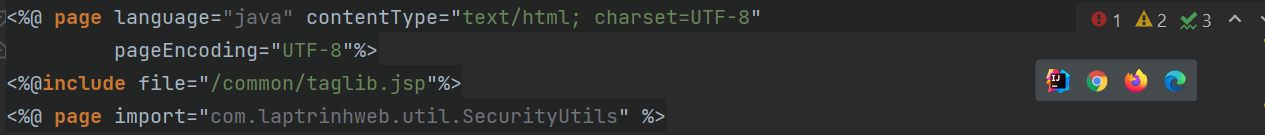
A screen shot of a computer code

Description automatically generated

Tiếp theo trong SecurityUtil, build một hàm getPrincipal để lấy ra thông tin chi tiết của user mà Spring Security đã lưu lại trong session từ phiên đăng nhập (Authication và Authorzation)



Để ở ngoài header lấy được tên đăng nhập thì thực hiện import và lấy ra các thuộc tính cần thiết



A screenshot of a computer program

Description automatically generated

Và khi thoát thì sẽ redirect về trang đăng nhập hoặc là trang home ở tình trạng chưa đăng nhập. Tiến hành tạo thêm controller ở home (controller cho việc logout)

A screen shot of a computer program

Description automatically generated

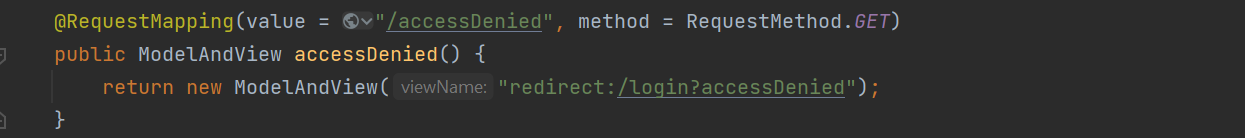
Và ở phần giao diện cũng phải gọi tới url logout để nó trỏ tới controller (nãy giờ là đang dùng controller của user để giới thiệu – admin cũng tương tự)



Lưu ý, để tránh trường hợp người dùng chưa đăng nhập nhưng vẫn muốn có đăng nhập vào trang chủ thì phải sử dụng intercept để ngăn ngừa việc này. Nó sẽ kiểm tra url đi vào trang chủ, sau đó xem xem role có phải ADMIN không (dỉ nhiên là không do người dùng kh đăng nhập). Nó sẽ tự động redirect tới trang đăng nhập (phần cấu hình này nằm trong file security.xml)



Còn một trường hợp là accessDined, nghĩa là user đã đăng nhập rồi, nhưng tài khoản này lại không có quyền ADMIN thì khi cố tình request url để vào trang chủ thì sẽ redirect tới trang login với trạng thái ?accessDenined



A screenshot of a login page

Description automatically generated

Đây là đoạn code sẽ thông báo ở phần giao diện (header.jsp của home)

A screen shot of a computer code

Description automatically generated