* Authentication (xác thực): là 1 quá trình yều cầu người dùng cung cấp thông tin đăng nhập hợp lệ để đăng nhập vào ứng dụng.
* Authorization (ủy quyền): là 1 quá trình xác nhận quyền mà người đăng nhập vào hệ thống có cho ứng dụng, dựa trên đó người dùng sẽ được chấp nhận truy cập vào các chức năng khác nhau hoặc quyền kiểm soát khác nhau của ứng dụng.
* Cấu hình Spring Security:
* B1. Định nghĩa file cấu hình security , namespace

<**beans:beans xmlns="http://www.springframework.org/schema/security"  
 xmlns:beans="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  
 http://www.springframework.org/schema/security  
 http://www.springframework.org/schema/security/spring-security-3.2.xsd"**>

…

* B2. Cấu hình web.xml

1. Cấu hình DelegatingFilterProxy trong web.xml

Trước khi hạn chế truy cập người dùng với đặc quyền nào đó, phải có 1 cách để biết ai đang sử dụng ứng dụng. Ứng dụng cần được xác nhận người dùng, yêu cầu họ đăng nhập vào hệ thống.

Để kích hoạt Spring Security cho ứng dụng web, phải thêm vào Filter như dưới cho web.xml. Trách nhiệm của Spring Security Filter là ngăn chặn 1 số URL để áp dụng xác nhận và ủy quyền như cấu hình trong file cấu hình security.

<**filter**>  
 <**filter-name**>springSecurityFilterChain</**filter-name**>  
 <**filter-class**>org.springframework.web.filter.DelegatingFilterProxy</**filter-class**>  
</**filter**>

<**filter-mapping**>  
 <**filter-name**>springSecurityFilterChain</**filter-name**>  
 <**url-pattern**>/\*</**url-pattern**>  
</**filter-mapping**>

1. Thêm địa chỉ file cấu hình security

<**context-param**>  
 <**param-name**>contextConfigLocation</**param-name**>  
 <**param-value**>  
 /WEB-INF/spring-security.xml   
 </**param-value**>  
</**context-param**>

* B3. Thêm chi tiết file cấu hình spring-secutiry.xml

<**http auto-config="true"**>  
 <**intercept-url pattern="/student/\*\*" access="ROLE\_USER"**/>  
</**http**>  
  
<**authentication-manager**>  
 <**authentication-provider**>  
 <**user-service**>  
 <**user name="hoang" password="123456" authorities="ROLE\_USER"**/>  
 </**user-service**>  
 </**authentication-provider**>  
</**authentication-manager**>

Thẻ <http> chỉ ra rằng chúng ta muốn sử dụng web security. Web security được cấu hình sử dụng <http>.

3 dòng XML đầu cấu hình Spring security để chặn các yêu cầu với “student” URL định dạng thông qua <intercept-url> và hạn chế truy cập chỉ cho những người dùng có quyền “ROLE\_USER”. Có thể sử dụng nhiều thẻ <intercept-url> để định nghĩa những yêu cầu truy cập khác nhau cho các URL khác nhau.

Đặt **auto-config=”true”** cung cấp cho ứng dụng 1 trang đăng nhập, hỗ trợ cho xác nhận HTTP cơ bản và hỗ trợ cho đăng xuất khỏi ứng dụng.

<authentication-manager> được sử dụng để định nghĩa 1 danh sách của các người dùng và mật khẩu ở bộ nhớ trong và thông tin quyền đăng nhập.

* Đăng nhập bằng form:

<**http auto-config="true" use-expressions="true"**>  
 <**intercept-url pattern="/student/\*\*" access="hasRole('ROLE\_USER)"** />  
  
 <**form-login login-page="/login" default-target-url="/student"  
 authentication-failure-url="/loginError"** />  
 <**logout logout-success-url="/logout" logout-url="/j\_spring\_security\_logout"** />  
 <**csrf** />  
</**http**>

Ở cấu hình trên sử dụng thành phần <form-login>. Nếu 1 user cố gắng truy cập bất cứ địa chỉ cấp quyền nào thì user sẽ được xác nhận với form login dựa trên cấu hình của <form-login>.

* Login-page: chỉ rõ đường dẫn của controller trả về trang đăng nhập. Ở đây URL là /login.
* Default-target-url: chỉ rõ đường dẫn của 1 phương thức trong controller mà sẽ đi đến sau khi xác nhận thành công, ở đây là /student
* Authentication-failure-url: chỉ rõ đường dẫn của phương thức trong controller nơi sẽ đi tới nếu xác nhận thất bại, ở đây là /loginError.