

- Annotation `@Configuration` xác định lớp `WebSecurityConfig` của ta là một lớp dùng để cấu hình.
- Annotation `@EnableWebSecurity` sẽ kích hoạt việc tích hợp Spring Security với Spring MVC.
- Trong lớp `WebSecurityConfig`, ta cần phải gọi đến interface `UserDetailsService` để cấu hình. Do đó ta sẽ inject `UserDetailsService`.
- Trong Spring Security, việc mã hóa mật khẩu sẽ do interface `PasswordEncoder` đảm nhận. `PasswordEncoder` có implementation là `BCryptPasswordEncoder` sẽ giúp chúng ta mã hóa mật khẩu bằng thuật toán BCrypt. Nhưng để sử dụng được `PasswordEncoder`, ta phải cấu hình để `PasswordEncoder` trở thành một **Bean**.
- Trong phương thức `configure(HttpSecurity http)`, ta sẽ cấu hình các chi tiết về bảo mật:

Phân quyền request

Trong đoạn code:

```
.authorizeRequests()
    .antMatchers("/register").permitAll()
    .antMatchers("/").hasRole("MEMBER")
    .antMatchers("/admin").hasRole("ADMIN")
    .and()
```

`antMatchers()` : khai báo đường dẫn của request

`permitAll()` : cho phép tất cả các user đều được phép truy cập.

`hasRole(roleName)` : chỉ cho phép các user có GrantedAuthority là `ROLE_roleName` mới được phép truy cập

Các bạn có thể tham khảo thêm các phương thức khác [tại đây](#)

Đăng nhập

Trong đoạn code:

```
.formLogin()
    .loginPage("/login")
    .usernameParameter("email")
    .passwordParameter("password")
    .defaultSuccessUrl("/")
    .failureUrl("/login?error")
    .and()
```

`loginPage()` : đường dẫn tới trang chứa form đăng nhập

`usernameParameter()` và `passwordParameter()` : giá trị của thuộc tính `name` của 2 input nhập username và password

`defaultSuccessUrl()` : đường dẫn tới trang đăng nhập thành công

`failureUrl()` : đường dẫn tới trang đăng nhập thất bại

Trong Spring Security, trang xử lý submit form mặc định là `/login`. Nếu bạn muốn custom thì có thể dùng `loginProcessingUrl()`.

Đăng xuất

Trong Spring Security, mặc định trang đăng xuất có đường dẫn là `/logout`. Sau khi đăng xuất, sẽ redirect về trang `/login?logout`. Ở đây mình sẽ giữ nguyên cấu hình mặc định. Nếu các bạn muốn custom thì có thể tham khảo [tại đây](#)