**Báo cáo Lab5 + Lab6**

1. **Lab5**
2. **Spring Boot REST API**

* **@RestController:** Là một controller, tất cả phương thức mapping vào trong controller này đều trở thành các operation, các operation có những mapping khác nhau:
  + **GET**(url): Collection<Student>

(Truy xuất) Nhận về tập hợp các sinh viên với địa chỉ url

* + **GET**(url): Student

(Truy xuất) Nhận về một sinh viên

* + **POST**(url,data): Student

(Tạo mới) Nhận về một sinh viên mới

* + **PUT**(url,data): Student

(Cập nhật) Nhận lại sinh viên với thông tin mới cập nhật

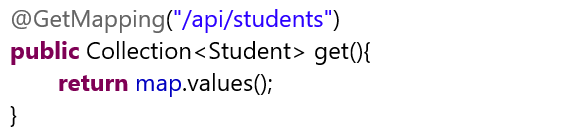
* + **DELETE**(url)

(Xóa) Xóa sinh viên (không trả dữ liệu về)

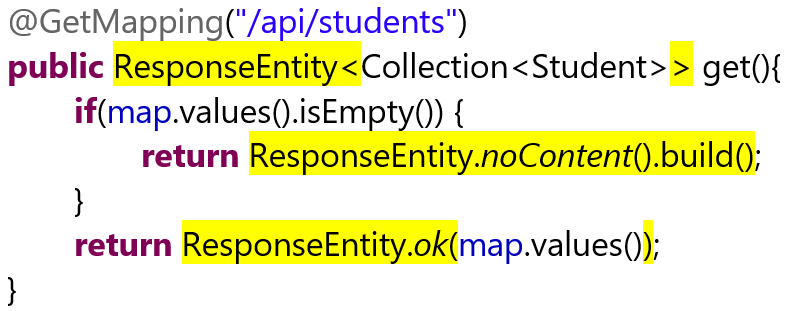
* Mặc định chỉ có các Rest Consumer cùng domain được cho phép consume các REST API (khác domain sẽ không được phép).
* **@CrossOrigin()** được sử dụng để khai báo cho phép các nguồn địa chỉ Rest Consumer đáng tin cậy.
* Cấu hình này cho phép các trang web đặt tại các host: localhost:8080 và 127.0.0.1:8080 được phép truy cập.
* Sử dụng origins = “\*” để cho phép mọi host
* **@ReponseEntity API:**

Có 2 cách viết mã, tuy nhiên cách viết 2 cho chúng ta mở rộng mã để điều khiển các lỗi một cách chính xác:

* + Cách 1:



* + Cách 2:



*(\*Rest Consumer sẽ nhận được trạng thái với mã 204. Từ đó có thể đưa ra các xử lý, thông báo phù hợp)*

* ResponseEntity.**badRequest**().build()

**400** Bad Request: Địa chỉ tồi

* ResponseEntity.**noContent**().build()

**204** No Content: Không có nội dung

* ResponseEntity.**notFound**().build()

**404** Not Found: Không tìm thấy

* ResponseEntity.**ok**(body)

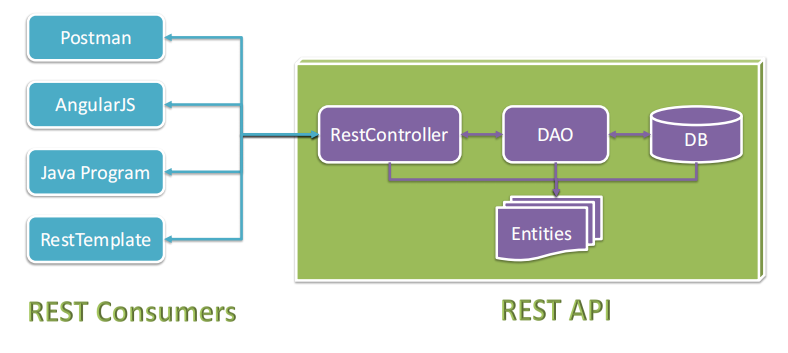
**200** OK: Thành công

* ResponseEntity.**status**(**HttpStatus**).build()

Status Code: Chứa trạng thái tùy chọn

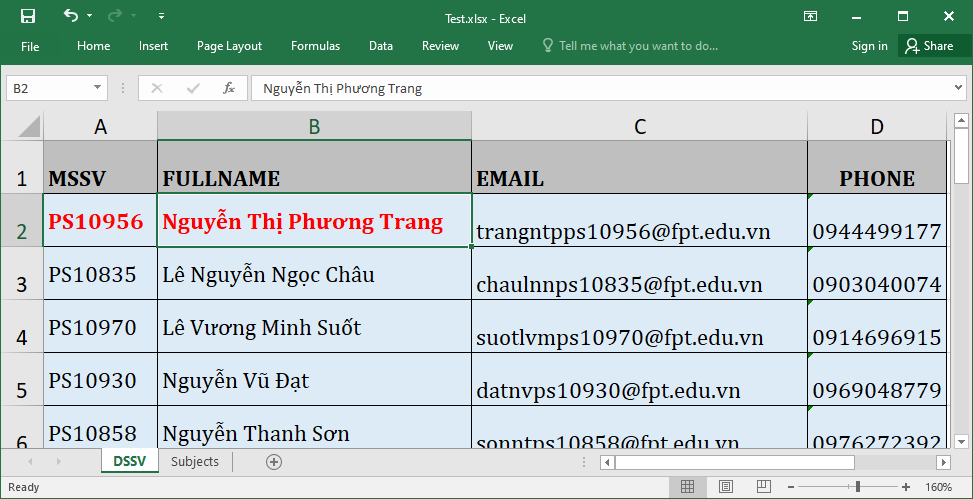
* **Return void:** Void là class đại diện cho void, được khai báo cho các phương thức không trả về kết quả

1. **REST API with JpaRepository**

****

* + **Xây dựng RestAPI:** Xây dựng DB -> Tạo Entities class -> Xây dựng lớp DAO -> Tạo RestController
  + **REST Consumers:** Xây dựng client để sử dụng RestAPI

1. **Import data Excel using REST API**



* + Sử dụng RestAPI để nhập dữ liệu từ excel vào cơ sở dữ liệu
  + Đọc dữ liệu từ file excel, mỗi row chuyển thành một đối tượng JSON và gửi lên server để lưu vào CSDL
  + API và thư viện cần thiết
    - **FileReader API:** đọc file từ trường file

FileReader được JS cung cấp sẵn, cho phép đọc dữ liệu file từ trường file để có thể xử lý ngay trên trình duyệt

* + - Thư viện **ExcelJS**: Xử lý dữ liệu excel của file

ExcelJS là thư viện JS cho phép xử lý dữ liệu từ file excel

* + Các thành phần trong ExcelJS
    - **Workbook**: gồm tất cả các sheet trong file excel
    - **Worksheet**: gồm nhiều hang (row)
    - **Row**: gồm nhiều ô (cell) theo hướng ngang
    - **Column**: gồm nhiều ô (cell) theo hướng đứng
    - **Cell**: chứa dữ liệu

1. **Upload/Download File REST API**
   * Tạo thư mục chứa file
   * Tạo FileManagerService (để quản lí file): Read(), Save()
   * Tạo RestController: FileManagerRestController

@PostMapping() dùng để upload

@GetMapping() dùng để download

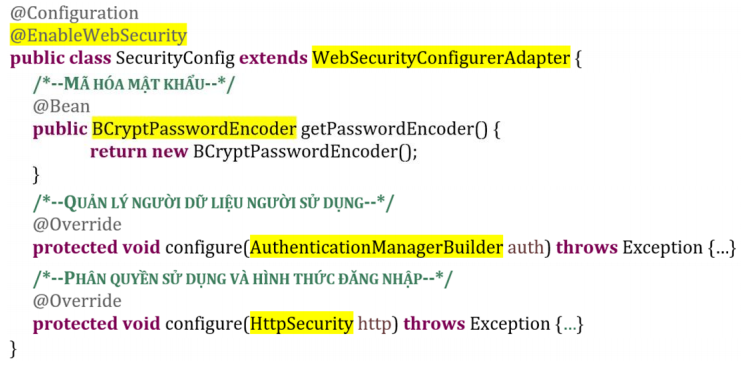
* + Rest Consumer:
    - Upload File (POST) đưa file lên server
    - Download File (GET) tải file từ server về client (lưu hoặc hiển thị)

*(ServletContext.****getRealPath****() để chuyển đổi đường dẫn ảo (đường dẫn tính từ gốc của website) sang đường dẫn thực.*

1. **Lab6**
2. **Web & Spring Security**
   * **Biểu hiện thường gặp của Web Security:** 
     + **Trang web trước và sau khi đăng nhập có thể khác nhau**
     + **Không thể thực hiện một số hành vi nếu chưa đăng nhập**
     + **Sau khi đã đăng nhập**
   * Giao diện có thể khác nhau tùy thuộc vào vai trò
   * Một số chức năng có thể thực hiện được hoặc không tùy vào vai trò
     + **Đăng nhập**
   * Từ trang web
   * Từ mạng xã hội
   * **Web Security concepts:**
     + **XSS – Cross-Site Scripting**
   * Ngăn chặn thực hiện của script từ data
     + **CORS – Cross-Site Resource Sharing**
   * Ngăn chặn chia sẻ tài nguyên
     + **CSRF – Cross-Site Request Forgery**
   * Ngăn chặn các request giả lập
     + **Authentication**
   * Ngăn chặn thực hiện khi chưa đăng nhập
     + **Authorization**
   * Ngăn chặn thực hiện khi đăng nhập không đúng vai trò
   * **Spring Security:**

* Spring Security là một framework, cung cấp các quy chuẩn trong việc thực hiện phòng vệ ứng dụng web
* Đơn giản trong Validation (XSS), CORS, CSRF
* Đa dạng hình thức phân quyền sử dụng
  + Cấu hình
  + @Annotation
  + Lập trình
  + Giao diện
* Chuẩn hóa mô hình dữ liệu người sử dụng và phương pháp mã hóa mật khẩu
* Tùy biến hình thức đăng nhập (Login UI), đăng xuất, xử lý lỗi truy cập không đúng vai trò
  + Đăng nhập với user từ mạng xã hội (facebook, gmail…)

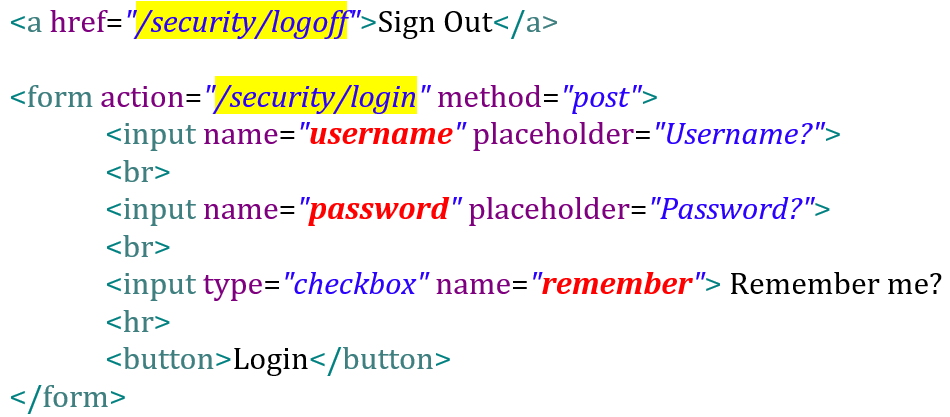
1. **Web Security case study**

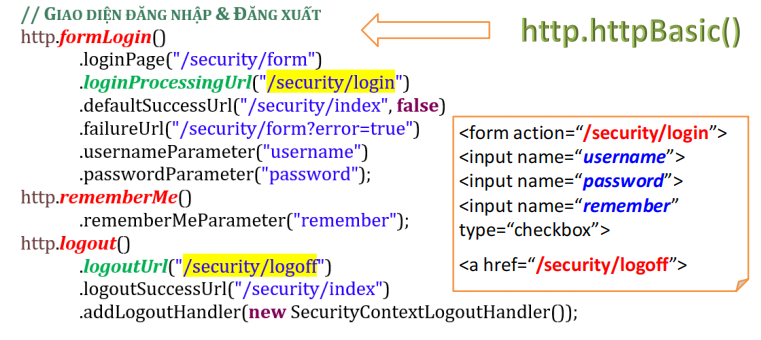
****

* Tạo file cấu hình: sử dụng annotation @EnableWebSecurity kế thừa từ WebSecurityConfigurerAdapter
* Mã hóa mật khẩu: sử dụng lớp BcryptPasswordEncoder
* Quản lý dữ liệu người sử dụng: Override phương thức configure nhận vào 1 đối số AuthenticationManagerBuilder (xây dựng nguồn dữ liệu người sử dụng)
* Phân quyền sử dụng và hình thức đăng nhập: Override phương thức configure nhận vào 1 đối số HttpSecurity (phân quyền sử dụng)

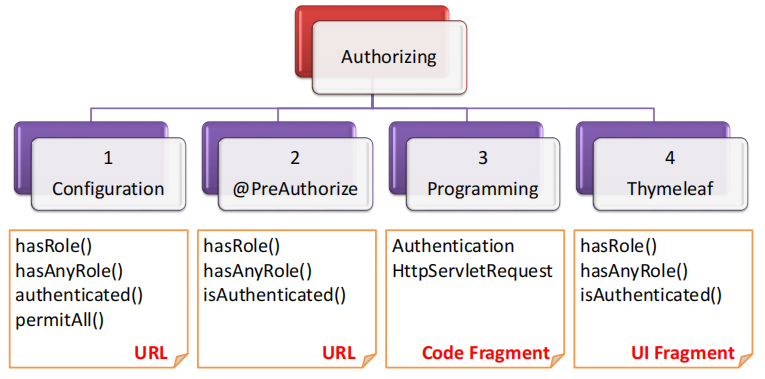
1. **HTTPSecurity – Login & Logoff**

* Thay thế hộp thoại đăng nhập bằng form HTML



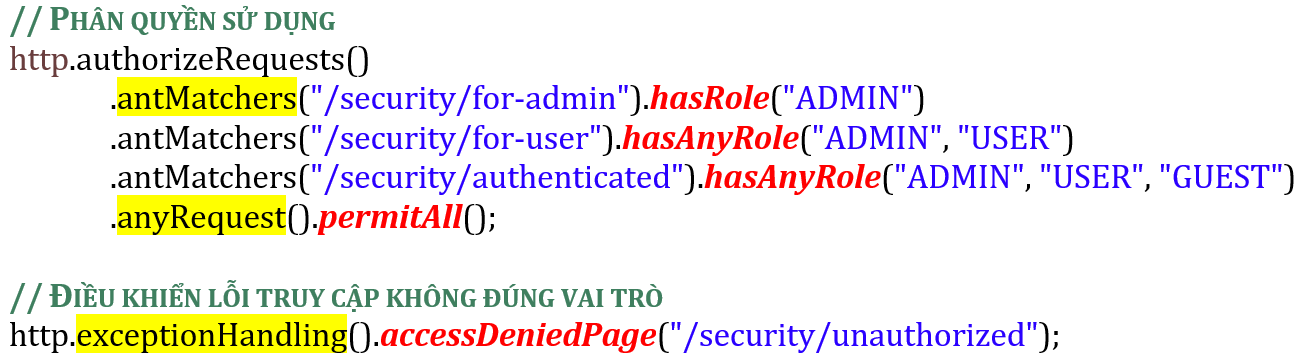


1. **HttpSecurity – Authorizing**

****

Trong Spring Boot có thể làm theo 4 cách: Configuration (cấu hình), @PreAuthorize (sử dụng annotation), Programming (lập trình), Thymeleaf (phân quyền trên giao diện)

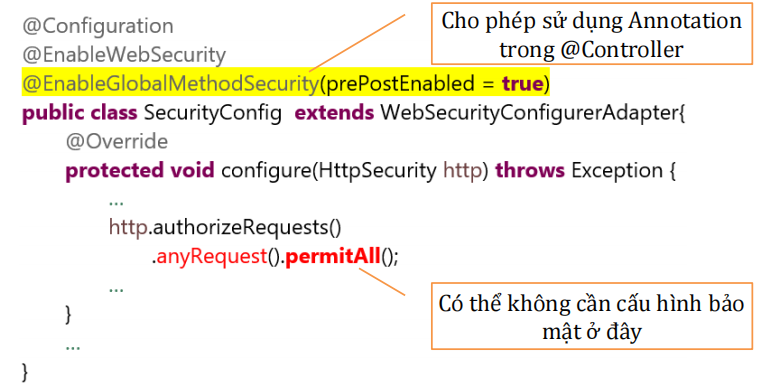
* Authorizing with configuration:



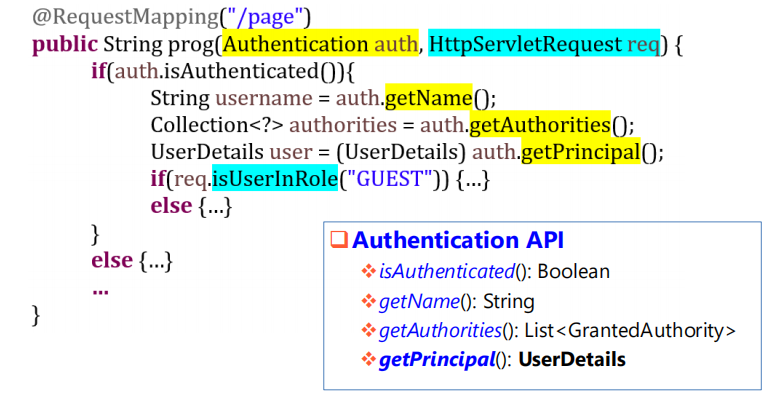
* XÁC ĐỊNH URLS
  + antMatchers(String…urlPatterns)
  + anyRequest()
* PHÂN QUYỀN
  + hasRole(String role) (phải được truy cập bằng vai trò Admin)
  + hasAnyRole(String…roles) (bất kể role nào trong những role này)
  + permitAll() (cho phép tất cả)
  + authenticated()
* Authorizing with @PreAuthorize(): đặt annotation lên trên các mapping



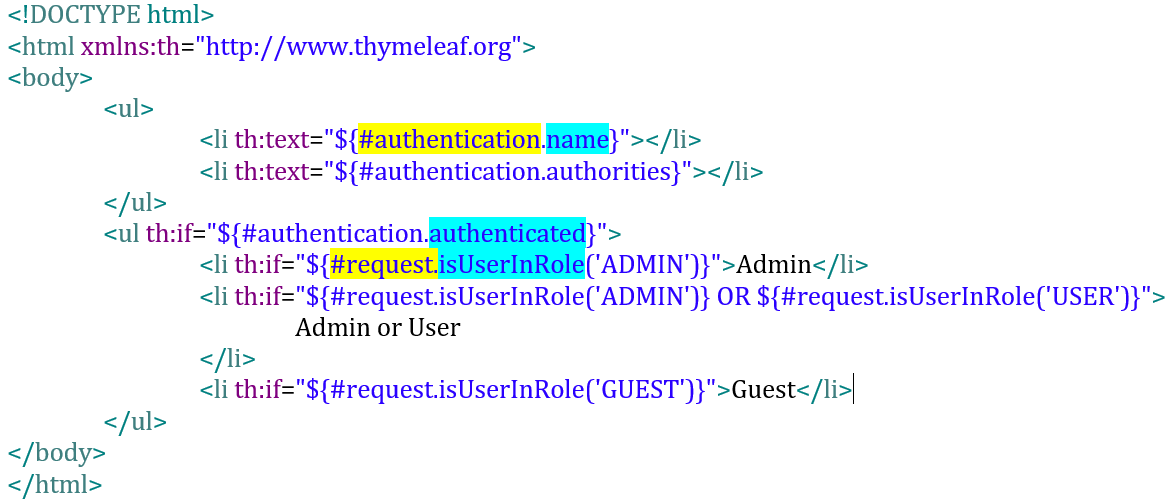
* + Phải thêm annotation



* Authorizing with Programming:



* Authorizing with thymeleaf: Có 2 cách
  + C1: Thymeleaf thuần túy: #Authentication, #request



* + C2: Thymeleaf Extras: sec:authentication, sec:authorize
    - Khai báo thư viện mở rộng

