Slot 9 – Spring Bot Security
Bảo vệ ứng dụng khỏi việc truy cập trái phép
- Xác thực (Authentication): Xác định người dùng là ai (user/pass/token,...)
- Phân quyền (Authorization): Người dùng đã xác thực được phép dùng những tào nguyên nào => cần có các vai trò: ROLE_USER; ROLE_ADMIN,….
====
Trong spring boot có khái niệm:
Filter Chain: Spring Security hoạt động qua chuỗi filter trước khi request đến controller => nghĩa là ứng dụng phải gọi đến FilterChain trước khi gọi đến Controller;
ví dụ về filterChain:
UsernamePasswordAuthenticationFilter
JwtAutheticationFilter
CsrfFilter
====
SecurityContext: lưu thông tin user đã xác thực (principal) trong SecurityContextHolder => giúp cho user không phải xác thực lại
========
Các loại xác thực (Authentication)
-User/Pass: truyền qua formLogin, Basic Auth
-Token-based: JWT, OAuth token
-Social Login: google , facebook
========

Authorization (phân quyền)
- ROLE_USER, ROLE_ADMIN
-Annotation: @PreAuthorize; @Secured
======
JWT: JSON WEB TOKEN
cấu trúc gồm 3 phần:
Header: dùng thuật toán (HS256, RS256,...)
Payload: thông tin user, vai trò, thời gian hết hạn
Signature: mã hóa header + payload bằng key bí mật

Cách dùng JWT
- sinh mã: mã hóa thông tin user + key bí mật
-Validate: decode token, check signature, check thời hạn
=> User login → server trả về JWT → Client lưu token (localStorage, session) → mỗi request gửi kèm Authorize → filter giải mã token và xác thực user
======
OAuth2: giao thức ủy quyền chuẩn, hay dùng để tích hợp với bên thứ 3 như google facebook
=> Client gửi user tới Authorization Server → user login vào server của bên thứ 3 → server trả code → client đổi lấy access token
VD: FAP
=====
CSRF Protection:

ngăn tấn công giả mạo yêu cầu POST từ trình duyệt mà user đã đăng nhập
=> form POST nên kèm CSRF Token để server xác thực hợp pháp

Khi nào cần disable CSRF???
- API restfull dùng JWT
- get request không thay đổi dữ liệu

=====