1. Spring Security

Spring Security là một framework bảo mật mạnh mẽ và linh hoạt được tích hợp trong Spring Boot. Nó cung cấp các tính năng và công cụ để xác thực và ủy quyền người dùng trong ứng dụng web. Spring Security cho phép xác định các quyền truy cập vào các phần của ứng dụng dựa trên vai trò của người dùng và các quy tắc bảo mật.

1.1. Xác thực (Authetication)

- Người dùng gửi thông tin đăng nhập (username và password) cho ứng dụng.
- Spring Security nhận thông tin đăng nhập và kiểm tra tính hợp lệ của thông tin đó.
- Spring Security sử dụng một Authentication Provider để kiểm tra thông tin đăng nhập và trả về một đối tượng Authentication nếu thông tin đúng.
- Đối tượng Authentication chứa thông tin về người dùng đã xác thực (ví dụ: username, password, vai trò...).

1.2. Phân quyền (Authorization)

- Sau khi xác thực thành công, Spring Security sử dụng đối tượng Authentication để kiểm tra quyền truy cập của người dùng vào các tài nguyên hoặc chức năng của ứng dụng.
- Các quyền truy cập được cấu hình thông qua các phương thức authorizeRequests() trong lớp cấu hình Security.

Ví dụ: .antMatchers("/admin/**").hasRole("ADMIN") chỉ cho phép người dùng có vai trò "ADMIN" truy cập vào các URL bắt đầu bằng "/admin/".

2. JWT

JSON Web Token (JWT) là một tiêu chuẩn mở được sử dụng để xác thực và phân quyền trong các ứng dụng web. JWT là một chuỗi JSON bao gồm các phần tử khác nhau: Header, Payload và Signature. JWT thường được sử dụng để xác thực người dùng và truyền thông tin giữa các bên một cách an toàn.

2.1. Cấu trúc của JSON Web Token:

Một JWT bao gồm ba phần chính: Header, Payload và Signature.

- Header: Phần Header chứa thông tin về loại token và thuật toán mã hóa được sử dụng để tạo chữ ký. Header thường được mã hóa dưới dạng Base64Url.
- Payload: Phần Payload chứa các thông tin có thể tùy chọn (claims) về người dùng và các metadata. Payload cũng được mã hóa dưới dạng Base64Url.
- Signature: Phần Signature được tạo bằng cách ký và mã hóa phần Header, phần Payload và một chìa khóa bí mật sử dụng thuật toán đã được chỉ định trong phần Header. Signature đảm bảo tính toàn vẹn của token và không cho phép bên thứ ba thay đổi nội dung.

2.2. Cách JWT hoạt động:

- Xác thực: Khi người dùng đăng nhập thành công vào một ứng dụng web, server sẽ tạo ra một JWT và gửi nó cho người dùng. JWT này chứa các thông tin xác thực về người dùng như ID, quyền hạn và thời gian hết hạn.
- Gửi thông tin: Sau khi người dùng đăng nhập và có JWT, họ có thể gửi nó kèm theo mỗi yêu cầu gửi đến server. Thông tin trong JWT cho phép server xác định danh tính của người dùng và kiểm tra xem họ có quyền truy cập vào tài nguyên yêu cầu hay không.
- Xác thực từ server: Khi server nhận được một yêu cầu kèm theo JWT, nó sẽ kiểm tra tính hợp lệ của JWT bằng cách giải mã và kiểm tra chữ ký. Nếu JWT hợp lệ, server sẽ thực hiện yêu cầu và trả về kết quả.

2.3 Lợi ích của JSON Web Token:

- Di động và dễ sử dụng: JWT dễ dàng tích hợp vào các ứng dụng di động và web, và nó có thể được truyền qua các giao thức như HTTP, WebSocket, ...
- Đơn giản và gọn nhẹ: JWT có cấu trúc đơn giản và kích thước nhỏ, giúp giảm thiểu lưu lượng mạng.
- Tính tự chứng minh (self-contained): JWT chứa tất cả thông tin cần thiết để xác thực và phân quyền, giúp giảm sự phụ thuộc vào cơ sở dữ liệu hoặc phiên trạng thái.
- Mở rộng được: JWT cho phép mở rộng bằng cách thêm các trường tùy chỉnh vào trong Payload.