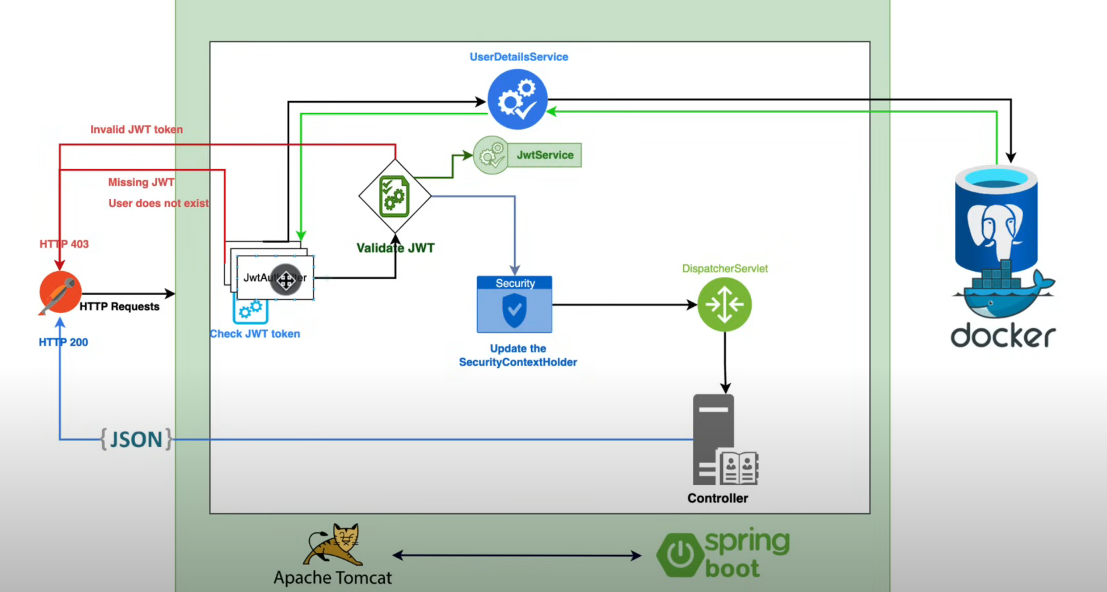
**Link tutorial:https://www.youtube.com/watch?v=KxqlJblhzfI**

**-Structure:**



**UserDetail**

-Khi trương trình spring security được chạy -> nó cần 1 người dùng để xác thực -> ta cần tạo 1 người dùng để xác thực và ta implement lớp người dùng đó với Inteface UserDetail của spring security -> tại sao lại cần triển khai UserDetail -> tại vì trong các phần khác của ứng dụng sẽ dùng cái userdetail này để xác thực.

**JWT Explain**

-1 mã JWT gồm 3 phần là : header,payload,signature.

Header: gồm 2 cặp khóa key value -> 1 cái là type thể hiện đây là loại gì jwt or … và 1 cái là algorithm là thuật toán dùng để mã hóa token đó.

Payload: chứa các cặp khóa key value mà người ta gọi đó là các claims chứa các thông tin như , thông tin người dùng, thời gian hết hạn của token…

Signature: Cái này nó được tạo bằng cách mã hóa 2 phần là header và payload bằng 1 secretkey.

**JWT Utils**

-Method extractAllClaims – trả về đối tượng Claims -> nhận vào 1 token -> Jwts.parserBuilder().setSigningKey() -> dùng secretKey đã được dùng để tạo token trước đó để giải mã. -> .build().parseClaimsJwt(token).getBody() -> lấy phần payload trong token -> trong payload là toàn bộ các claims -> lấy hết claims.

-Method extractClaim(String token,Funtion<Claims,T> claimsResolver)-> trả về claimsResolver.apply(claims).

-Method extractUsername -> trả về username -> sử dụn hàm extractClaim để lấy -> Subject là nơi lưu username.

-Method generateToken

-Method isValidToken

**ApplicationConfig**

-Lớp Config này dùng để định nghĩa lại cách hoạt động của các interface UserDetailsService , PasswordEncoder,AuthenticationProvider -> lớp giao tiếp với cơ sở dữ liệu định nghĩa lớp nào được dùng để xác thực người dùng và cách mã hóa mật khẩu người dùng.

**JwtAuthentication -> Implement OncePerRequestFilter**

-Đầu tiên khi người dùng gửi 1 request đến phía server thì nó sẽ phải đi qua Lớp JwtAuthenticationFilter để thực hiện xác thực như username and password hoặc token hết hạn … Và để làm điều này ta cần implement 1 interface OncePerRequestFilter.

1.Điều đầu tiền cần làm trong lớp overwrite của OncePerRequestFilter là kiểm tra token -> nó null hay không xuất phát với ký tự “Bear ” ở đầu.

2.Lấy username từ Token của người dùng -> check có tên của username trong subject không.

3.Check user từ Token đó xem có tồn tại trong hệ thống không -> trọc vào DB sử dụng UserDetailService -> trả ra UserDetails.

4.Sử dụng token và cái Thằng UserDetails đó và gọi hàm isValidToken trong jwtUtils và chuyền nó vào hàm đó để check valid.

5.Thêm vào SecurityContextHolder.

6.Chạy những Filter Tiếp theo.

**SecurityContextHolder**

-Lưu trữ người dùng đã được xác thực và các FilterChain dựa vào người dùng được chứa trong SecurityContextHolder để thực hiện xác thực người dùng đó có quyền truy cập vào các tài nguyên không.