라이브 세션-Spring Security 기본-2022.09.27(금)

✅ Session vs Token 비교

1️⃣ Session 기반 자격 증명 방식

- 세션은 인증된 사용자 정보를 **서버 측 세션 저장소에서 관리한다**.
- **세션 ID**는 **클라이언트의 쿠키에 저장**되어 request 전송 시, 인증된 사용자인지를 증명하는 수단으로 사용된다.
- 세션 ID만 클라이언트 쪽에서 사용하므로 **상대적으로 적은 네트워크 트래픽을 사용**합니다.
- 서버 측에서 세션 정보를 관리하므로 **보안성 측면에서 조금 더 유리**합니다.
- 서버의 확장성 면에서는 **세션 불일치 문제가 발생**할 가능성이 높습니다.
- 세션 데이터가 많아지면 질수록 **서버의 부담이 가중**될 수 있습니다.
- **SSR(Server Side Rendering) 방식의 애플리케이션에 적합한 방식**입니다.

2️⃣ 토큰 기반 자격 증명 방식

- 토큰에 포함된 인증된 사용자 정보는 서버 측에서 별도의 관리를 하지 않습니다.
- 생성된 토큰을 헤더에 포함시켜 request 전송 시, 인증된 사용자인지를 증명하는 수단으로 사용된다.
- 토큰내에 인증된 사용자 정보 등을 포함하고 있으므로 **세션에 비해 상대적으로 많은 네트워크 트래픽을 사용한다.**
- 기본적으로 서버 측에서 토큰을 관리하지 않으므로 보안성 측면에서 조금 더 불리하다.
- 인증된 사용자 request의 상태를 유지할 필요가 없기 때문에 **서버의 확장성 면에서 유리하고, 세션 불일치 같은 문제가 발생하지 않는다.**

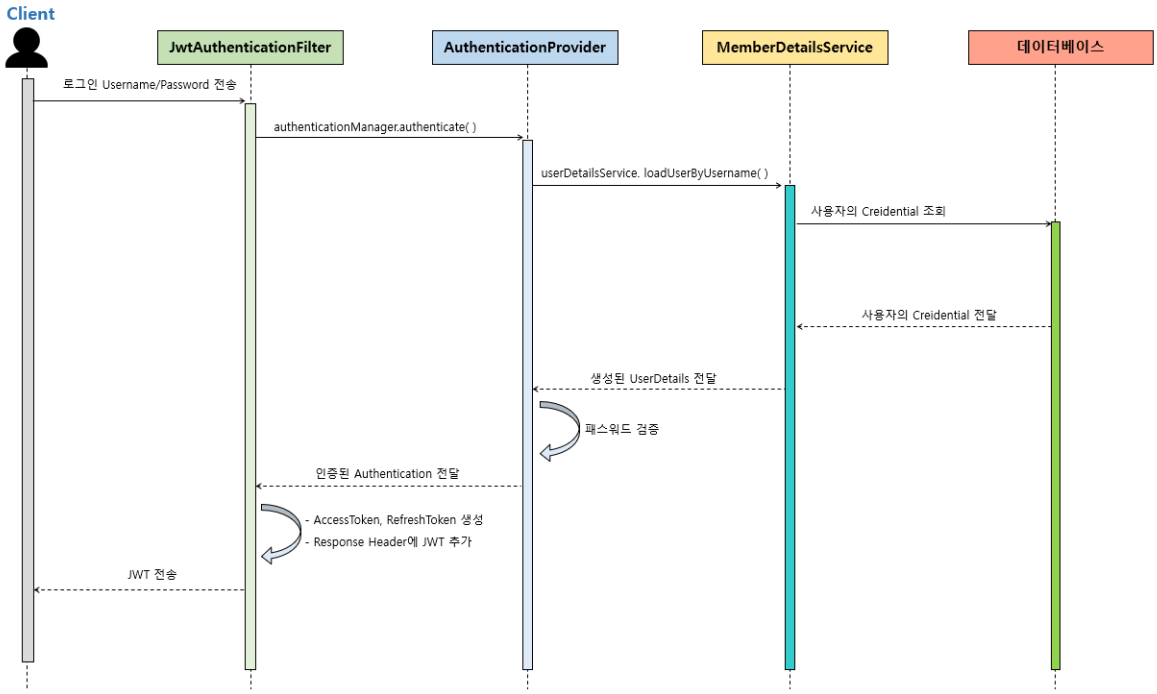
- 토큰에 포함되는 사용자 정보는 토큰의 특성상 암호화가 되지 않기때문에 공격자에게 토큰이 탈취될 경우, 사용자 정보를 그대로 제공하는 셈이됩니다. 따라서 **민감한 정보는 토큰에 포함시키지 말아야 한다.**
- 기본적으로 토큰이 만료되기 전까지는 **토큰을 무효화 시킬 수 없다.**
- CSR(Server Side Rendering) 방식의 애플리케이션에 적합한 방식입니다.

✅ JWT(Json Web Token)란?

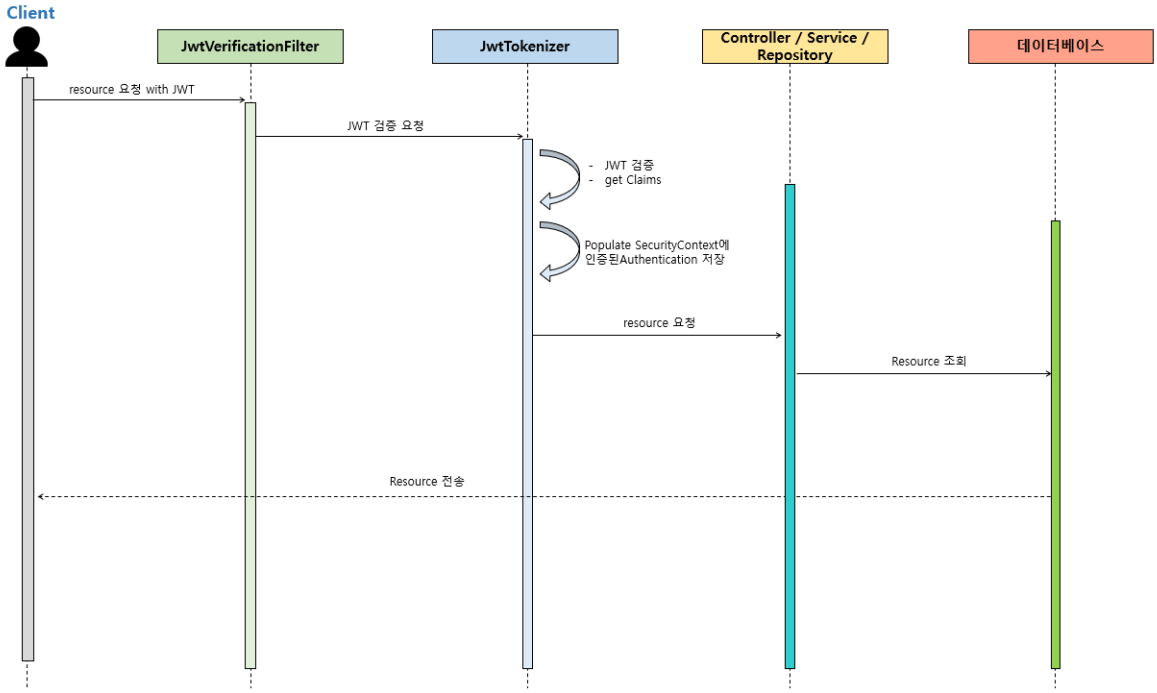
- 무상태(Stateless) 애플리케이션에서 인증된 사용자의 자격 증명을 위해 사용되는 토큰
- Access Token과 Refresh Token을 사용할 수 있다.
- Access Token은 자격 증명용
- Refresh Token은 Access Token 갱신용

✅ 커피 주문 샘플 애플리케이션에 JWT 적용

1️⃣ 로그인 인증 시, JWT 생성 흐름



2️⃣ JWT를 이용한 클라이언트의 자격 검증 처리 흐름



3️⃣ JwtTokenizer 리뷰

- 소스 코드로 리뷰

4️⃣ JwtAuthenticationFilter 리뷰

- 소스 코드로 리뷰

5️⃣ AuthenticationSuccessHandler / AuthenticationFailureHandler 리뷰

- 소스 코드로 리뷰

6️⃣ JwtVerificationFilter 리뷰

- 소스 코드로 리뷰

7️⃣ SecurityConfiguration 리뷰

- 소스 코드로 리뷰

8️⃣ AuthenticationEntryPoint

- 소스 코드로 리뷰

9️⃣ AccessDeniedHandler

- 소스 코드로 리뷰