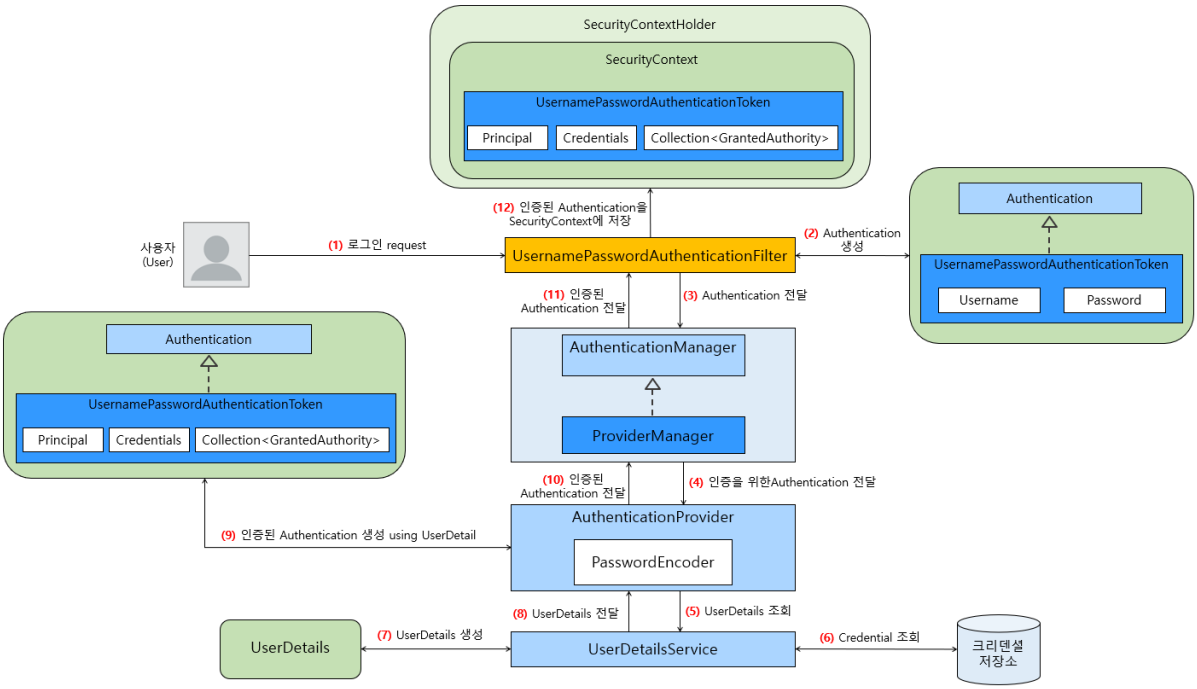
라이브 세션-Spring Security 기본-2022.09.23(금)

✅ Spring Security에서의 인증(Authentication) 처리 흐름



- `UsernamePasswordAuthenticationFilter` 는 Username/Password 기반의 인증 요청을 처리한다.
  - `UsernamePasswordAuthenticationToken` 생성
    - **아직 인증되지 않은 Authentication 객체**

- `AuthenticationManager`는 인증 처리를 총괄하는 매니저 역할을 하는 인터페이스.

- `ProviderManager` 는 `AuthenticationManager`를 구현한 구현 클래스
- `UserDetails` 는 사용자의 자격을 증명해주는 `크리덴셜(Credential)` 과 권한 정보를 가지고 있는 객체.
- `UserDetailsService` 가 사용자의 Credential과 권한 정보를 조회해서 `UserDetails` 를 생성.
- `UserDetailsService` 가 `AuthenticationProvider` 에게 `UserDetails` 를 전달.
- `AuthenticationProvider` 는 전달 받은 `UserDetails` 에서 패스워드가 일치하는지 검증.
- `AuthenticationProvider` 가 패스워드 검증에 성공하면 **인증에 성공한 사용자의 정보 (Principal, Credential, GrantedAuthorities)를 포함한 Authentication을 생성.**
- **인증된 Authentication**을 전달 받은 **UsernamePasswordAuthenticationFilter**는 `SecurityContextHolder`를 이용해 `SecurityContext` 에 **인증된 Authentication**을 저장.

  `SecurityContext`는 다시 **HttpSession** 에 저장되어 **사용자의 인증 상태를 유지.**

✅ Spring Security의 인증 컴포넌트

1️⃣ UsernamePasswordAuthenticationFilter extends AbstractAuthenticationProcessingFilter

2️⃣ UsernamePasswordAuthenticationToken

3️⃣ AuthenticationManager

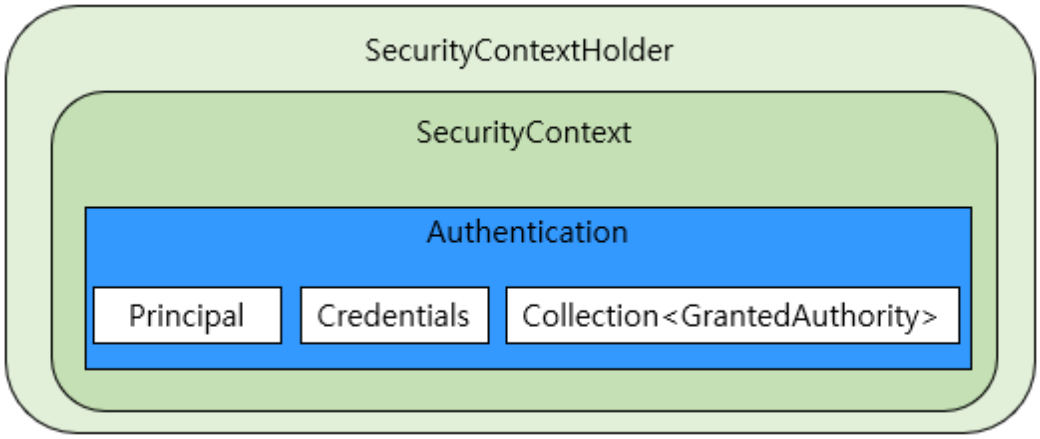
4️⃣ ProviderManager

5️⃣ AuthenticationProvider

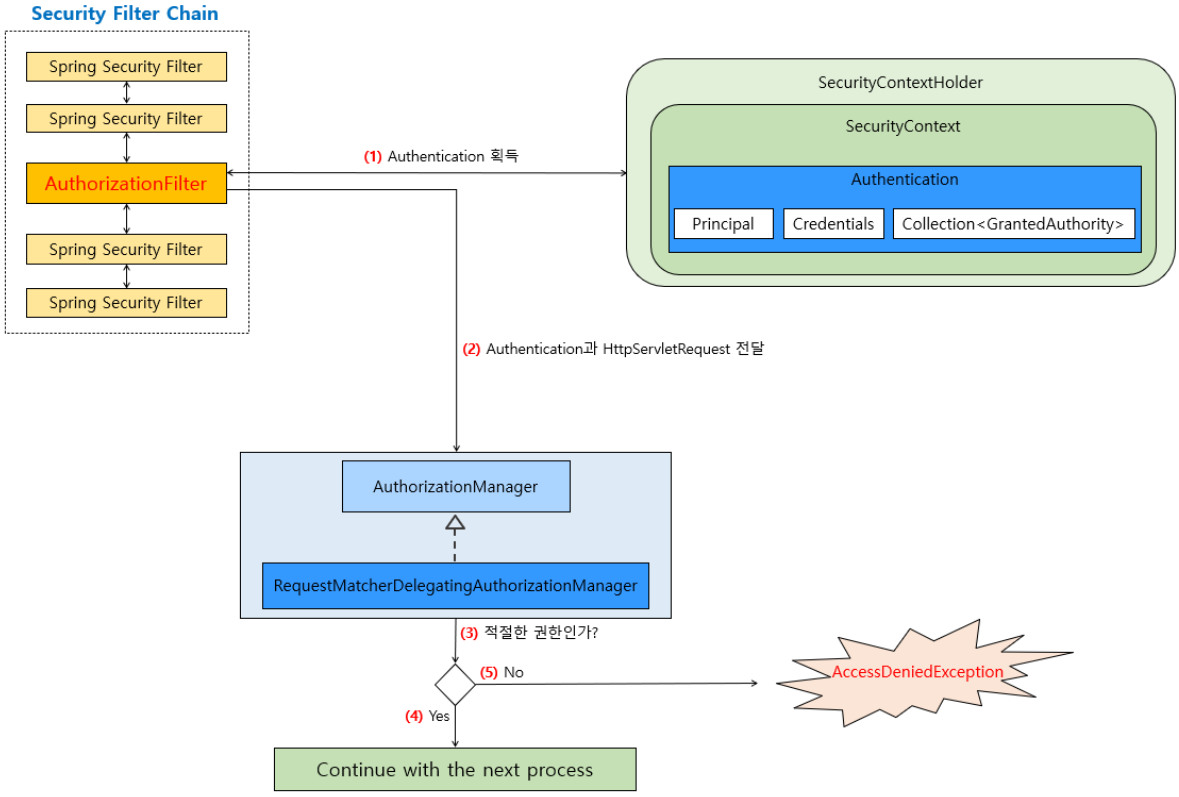
6️⃣ UserDetails

7️⃣ UserDetailsService

8️⃣ SecurityContext와 SecurityContextHolder



✅ Spring Security에서의 권한 부여 (Authorization) 처리 흐름



- Spring Security Filter Chain에서 URL을 통해 사용자의 액세스를 제한하는 **권한 부여 Filter**는 바로 `AuthorizationFilter` 이다.
    - SecurityContextHolder로 부터 Authentication을 획득
- SecurityContextHolder로 부터 획득한Authentication과 HttpServletRequest를 `AuthorizationManager` 에게 전달
- `AuthorizationManager` 는 권한 부여 처리를 총괄하는 매니저 역할을 한다.
- `RequestMatcherDelegatingAuthorizationManager` 는 `AuthorizationManager` 를 구현하는 구현체 중 하나이다.
    - `RequestMatcherDelegatingAuthorizationManager` 는 RequestMatcher 평가식을 기반으로 해당 평가식에 매치되는 `AuthorizationManager` 에게 권한 부여 처리를 위임하는 역할

    - `RequestMatcherDelegatingAuthorizationManager` 가 직접 권한 부여 처리를 하는 것이 아니라 `RequestMatcher`를 통해 매치되는 `AuthorizationManager` 구현 클래스에게 위임
    - `RequestMatcherDelegatingAuthorizationManager` 내부에서 매치되는 `AuthorizationManager` 구현 클래스가 있다면 해당 `AuthorizationManager` 구현 클래스가 사용자의 권한을 체크
- 적절한 권한이 아니라면 (5)와 같이 `AccessDeniedException`이 throw되고 **ExceptionTranslationFilter**가 `AccessDeniedException`을 처리

✅ Hello Spring Security 샘플 애플리케이션 리뷰(Optional)

- 코드로 확인