

Spring Security

SpringBoot 기준으로 서술

Java, Spring

Overview

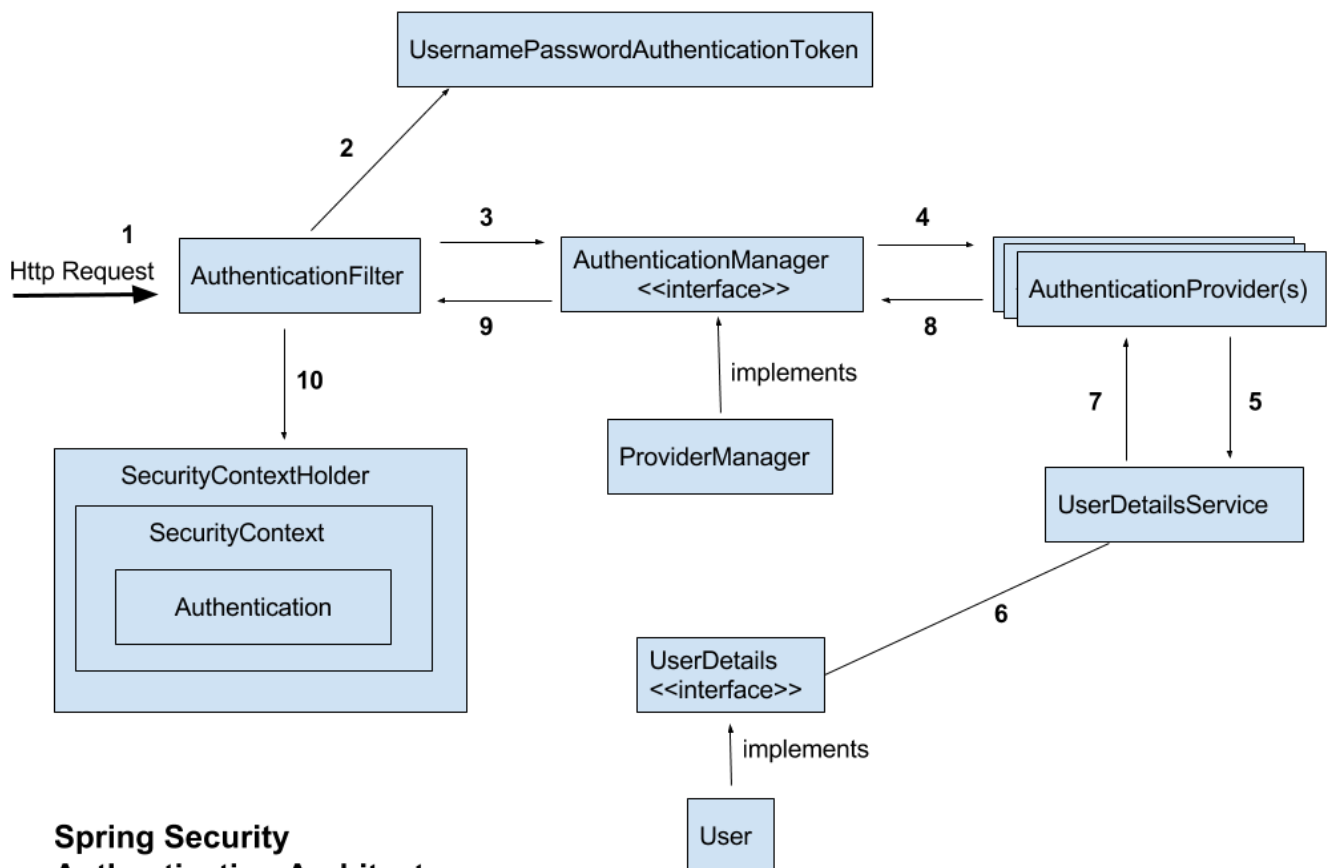
- Authentication : 인증. “유효한 사용자인가?”
- Authorization : 허가(권한 부여). 유효한 사용자가 이 작업을 할 수 있는가?

Documents

- [Spring Security Architecture](#)
- [Spring Security : Authentication Architecture](#)
- [\[Docs\] Spring Security - Servlet Applications](#)
- [Samples](#)

Authentication

Docs



Spring Security Authentication Architecture

Chathuranga Tennakoon
www.springbootdev.com

- 모두 Interface

```
classDiagram
    class AuthenticationManager {
        authenticate(Authentication authentication)
        Authentication
    }
    class AuthenticationProvider {
        authenticate(Authentication authentication)
        Authentication
        supports(Class~? authentication) boolean
    }
    class UserDetailsService {
        loadUserByUsername(String username) UserDetails
    }
    class UserDetails {
        getAuthorities() Collection~?
        extends GrantedAuthority~
        getPassword() String
        isAccountNonExpired() boolean
        isAccountNonLocked() boolean
        isCredentialsNonExpired() boolean
        isEnabled() boolean
    }
    AuthenticationManager --> AuthenticationProvider : authRequest
    AuthenticationProvider <..> AuthenticationProvider
    UserDetailsService : 서비스 주입 후 authenticate()에서 사용자 정보를 꺼내어 사용
    UserDetailsService -- UserDetails
```

AuthenticationManager

- AuthenctaionManager
 - 기본 구현체 : [ProviderManager](#)
 - [AuthenticationManagerBuilder](#) : [WebSecurityConfigurerAdapter](#) 의 `configure(AuthenticationManagerBuilder auth)` Method로 “auth” 인자를 사용할 수 있는데, 이 인자의 “userDetailsService(T userDetailsService)”로 “UserDetailsService” 주입이 가능함.

```
AuthenticationManager - org.springframework.security.authentication
AuthenticationManagerDelegator - org.springframework.security.config.annotation.authentication.configuration.AuthenticationConfiguration
AuthenticationManagerDelegator - org.springframework.security.config.method.GlobalMethodSecurityBeanDefinitionParser
AuthenticationManagerDelegator - org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter
NoOpAuthenticationManager - org.springframework.security.oauth2.client.filter.OAuth2ClientAuthenticationProcessingFilter
NoOpAuthenticationManager - org.springframework.security.access.intercept.AbstractSecurityInterceptor
OAuth2AuthenticationManager - org.springframework.security.oauth2.provider.authentication
ProviderManager - org.springframework.security.authentication
```

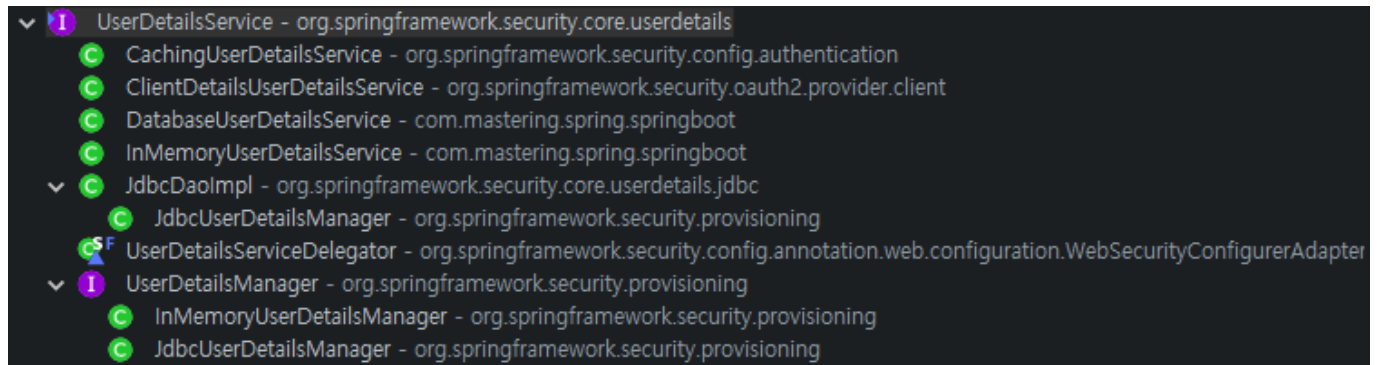
AuthenticationProvider

- AuthenticationProvider

```
AuthenticationProvider - org.springframework.security.authentication
AbstractJaasAuthenticationProvider - org.springframework.security.authentication.jaas
DefaultJaasAuthenticationProvider - org.springframework.security.authentication.jaas
JaasAuthenticationProvider - org.springframework.security.authentication.jaas
AbstractUserDetailsAuthenticationProvider - org.springframework.security.authentication.dao
DaoAuthenticationProvider - org.springframework.security.authentication.dao
AnonymousAuthenticationProvider - org.springframework.security.authentication
NullAuthenticationProvider - org.springframework.security.config.authentication.AuthenticationManagerBeanDefinitionParser
OidcAuthenticationRequestChecker - org.springframework.security.config.annotation.web.configurers.oauth2.client.OAuth2LoginConfigurer
PreAuthenticatedAuthenticationProvider - org.springframework.security.web.authentication.preauth
RememberMeAuthenticationProvider - org.springframework.security.authentication
RemoteAuthenticationProvider - org.springframework.security.authentication.rcp
RunAsImplAuthenticationProvider - org.springframework.security.access.intercept
TestingAuthenticationProvider - org.springframework.security.authentication
```

UserDetailsService

- UserDetailsService
 - UserDetails



Authorization

[Docs](#)

Protection Against Exploits

[Docs](#)

From:

<http://wiki.ledyx.xyz/> - **Ledyx WIKI**

Permanent link:

<http://wiki.ledyx.xyz/doku.php?id=back-end:spring:security&rev=1619696834>

Last update: **2021/04/29 11:47**

