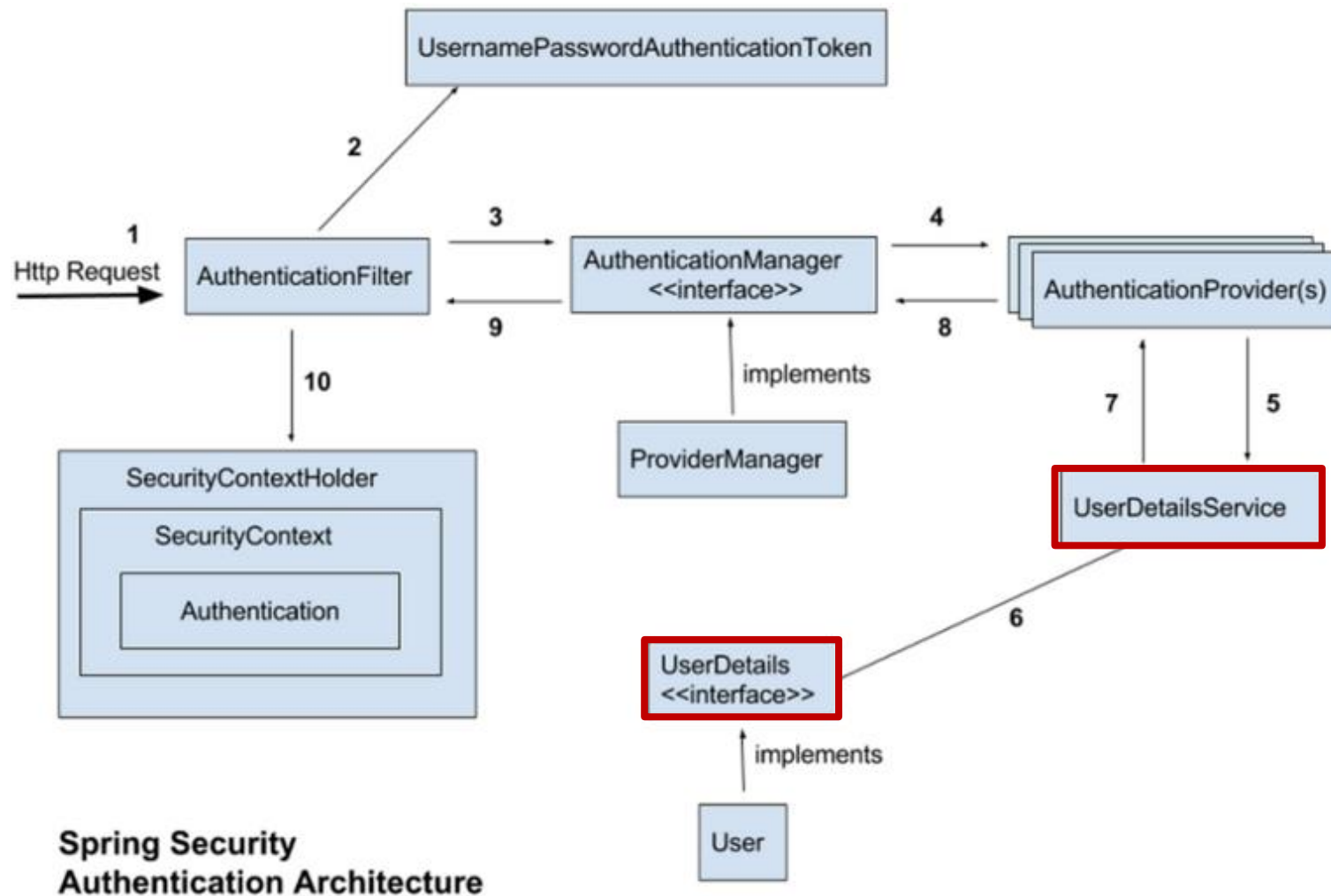


2024년 상반기 K-디지털 트레이닝

Spring Web Security 소개

[KB] IT's Your Life

✓ 스프링 시큐리티 아키텍처



✓ 프로젝트 생성

- Template: MyBatisSpringLegacy
- name: secserver

✓ build.gradle

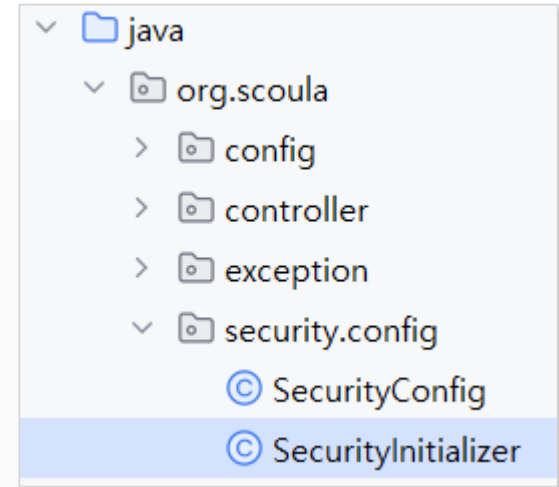
```
...
ext {
    junitVersion = '5.9.2'
    springVersion = '5.3.37'
    ...
    springSecurityVersion='5.8.13'
}

dependencies {
    ...

    // 보안
    implementation("org.springframework.security:spring-security-web:${springSecurityVersion}")
    implementation("org.springframework.security:spring-security-config:${springSecurityVersion}")
    implementation("org.springframework.security:spring-security-core:${springSecurityVersion}")
    implementation("org.springframework.security:spring-security-taglibs:${springSecurityVersion}")
    ...
}
```

✓ SecurityInitializer.java

```
package org.scoula.security.config;  
  
import org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer;  
  
public class SecurityInitializer extends AbstractSecurityWebApplicationInitializer {  
  
}
```

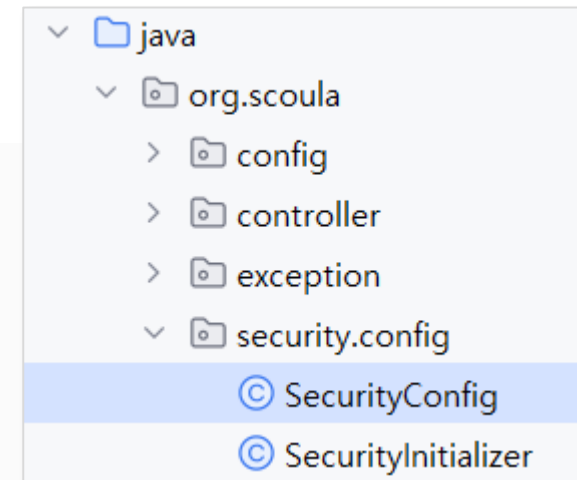


✓ SecurityConfig.java

```
package org.scoula.security.config;

@Configuration
@EnableWebSecurity
@Log4j
public class SecurityConfig extends WebSecurityConfigurerAdapter {

}
```



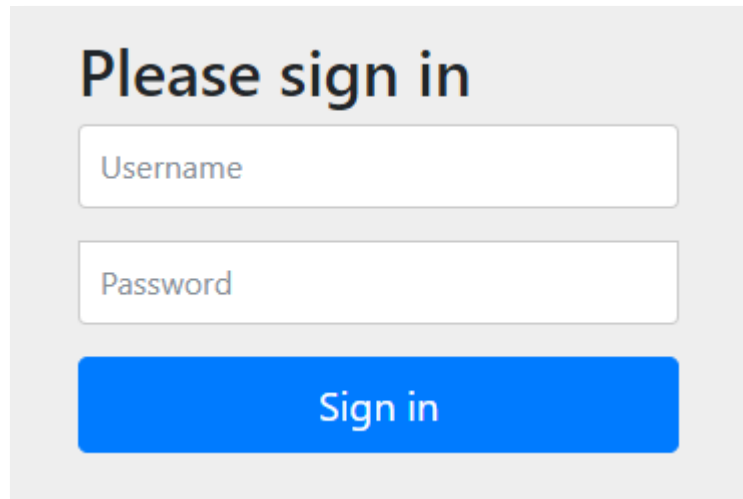
✓ config/WebConfig.java

```
@Override
public Class<?>[] getRootConfigClasses() {
    return new Class[] { RootConfig.class, SecurityConfig.class };
}
```

✓ 실행

- 모든 페이지는 보안정책에 따라 접근
- 스프링 시큐리티가 설정되면
 - 디폴트 : 모든 페이지는 로그인을 해야 접근 가능
 - /login 페이지로 리다이렉트됨

<http://localhost:8080/login>

A screenshot of a web browser showing a login page. The page has a light gray background. At the top, it says "Please sign in" in a bold, dark font. Below this, there are two input fields: the first is labeled "Username" and the second is labeled "Password". Both fields are white with a light gray border. Below the input fields is a blue button with the text "Sign in" in white. The entire form is centered on the page.

Please sign in

Username

Password

Sign in

✓ 인증

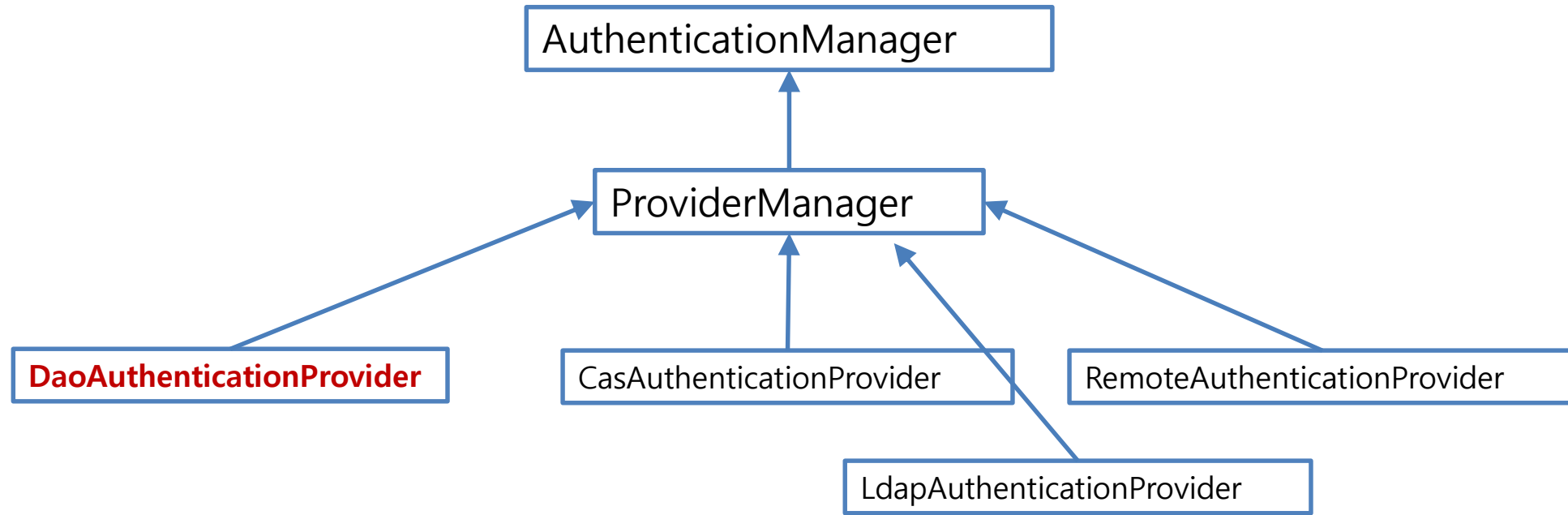
- 자신을 증명하는 것
- 자기 스스로가 무언가 자신을 증명할 만한 자료를 제시

✓ 인가

- 남에 의해서 자격이 부여됨

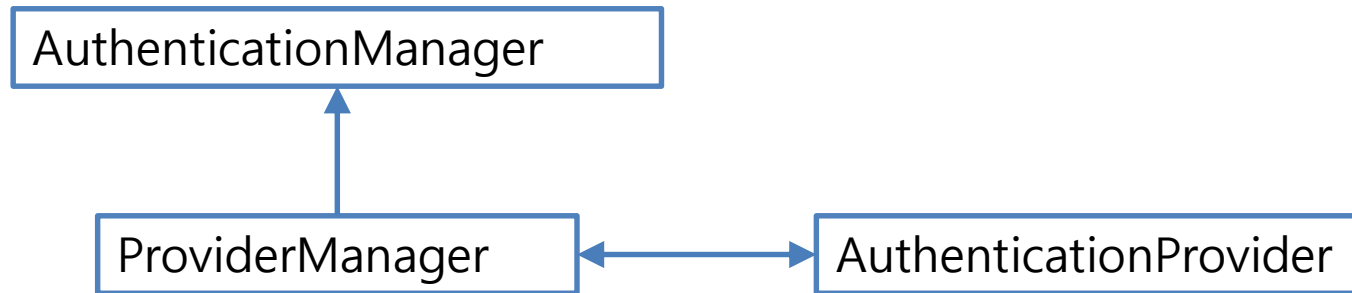
✓ AuthenticationManager(인증 매니저)

- 인증 담당하며, 다양한 방식의 인증을 처리



✓ ProviderManager

- 인증에 대한 처리를 AuthenticationProvider라는 타입의 객체를 이용해 처리



✓ AuthenticationProvider(인증 제공자)

- 실제 인증 작업을 진행
- UserDetailsService가 인증된 정보에 권한에 대한 정보를 구성



- ✓ 스프링 시큐리티를 커스터마이징 하는 방식
 - AuthenticationProvider를 직접 구현하는 방식
 - 실제 처리를 담당하는 UserDetailsService를 구현하는 방식