

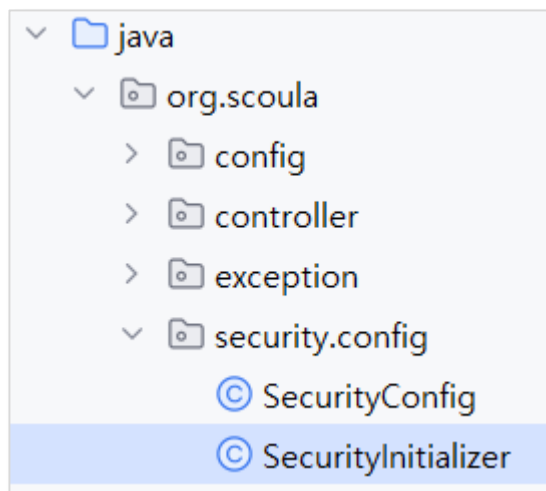
2025년 상반기 K-디지털 트레이닝

Spring Web Security 소개

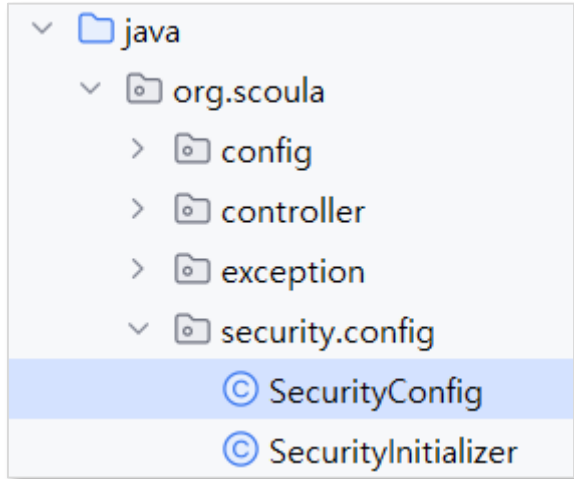
[KB] IT's Your Life

- ✓ 다음과 같이 프로젝트를 생성하고, Spring Security 의존성을 추가하세요.
 - Template: MyBatisSpringLegacy
 - name: secserver

- ✔ 다음과 같이 SecurityInitializer 클래스를 추가하고, 기본 코드를 작성하세요.
 - AbstractSecurityWebApplicationInitializer 상속



- ✓ 다음과 같이 SecurityConfig 클래스를 추가하고, 기본 골격을 작성하세요.

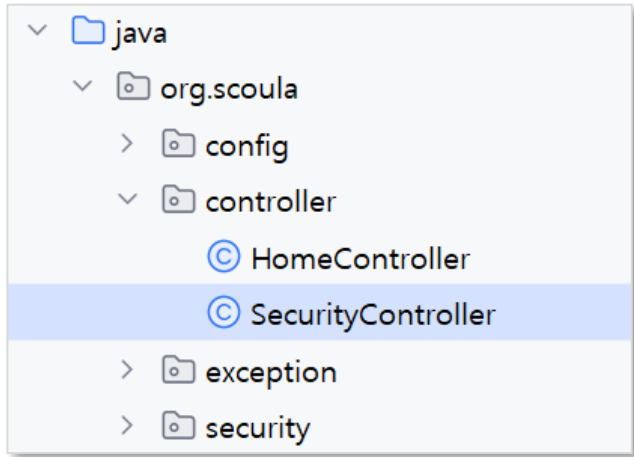


- ✓ WebConfig에 SecurityConfig 설정을 추가하세요.

- ✓ **http://localhost:8080/ 접속시 로그인 페이지로 리다이렉트되는지 확인하세요.**

- ✔ SecurityConfig에서 UTF-8 문자셋 인코딩 필터를 작성하고 CSRF 필터 앞에 배치하세요.

- ✔ 다음과 같이 SecurityController를 추가하고, 다음을 처리하도록 작성하세요.



- /security/all 접속시 로그 출력 및 sercurity/all 뷰로 이동
- /security/member 접속시 로그 출력 및 sercurity/member 뷰로 이동
- /security/admin 접속시 로그 출력 및 sercurity/admin 뷰로 이동
- all, member, admin 뷰 jsp 작성

- ✔ SecurityConfig에서 다음 경로에 대해서 접근 권한을 설정하세요.
 - /security/all : 모두 접근 가능
 - /security/member: ROLE_MEMBER 권한 이상 인 경우 접근 가능
 - /security/admin: ROLE_ADMIN 권한인 경우 접근 가능

- ✔ 로그인 안한 상태에서 접근 권한이 없는 경우 로그인 페이지로 이동하도록 설정하세요.

✓ 다음과 같은 메모리 기반의 사용자 정보를 등록하세요.

- username: admin
- password: 1234(암호화 없음)
- 권한: ROLE_ADMIN, ROLE_MEMBER

- username: member
- password: 1234(암호화 없음)
- 권한: ROLE_MEMBER

- ✔ SecurityController에 /security/login 요청을 처리하는 메서드와 뷰를 작성하고, 로그인 페이지를 자신이 작성한 페이지로 운영하세요.

✓ 다음과 같이 로그아웃 처리되도록 설정하세요.

- 로그아웃 요청 POST URL: /security/logout
- 로그아웃 성공시 리다이렉트 URL: /security/logout
- SecurityController에 /security/logout GET 요청처리 메서드 추가
- security/logout 뷰 작성

- ✓ member.jsp, admin.jsp 페이지에 로그아웃 폼을 추가하고, 실제 로그아웃 되는지 확인하세요.

- ✓ PasswordEncoder 빈을 SecurityConfig에 등록하고, 단위 테스트를 이용하여 비밀번호 암호화 및 검증을 테스트하세요.

- ✓ 메모리 등록 유저의 비밀번호를 암호화된 비밀번호로 대체하세요.