

폼기반!!

2025년 상반기 K-디지털 트레이닝

SecurityConfig 전체 코드

[KB] IT's Your Life

SecurityConfig.java

```
package org.scoula.security.config;
...
@Configuration
@EnableWebSecurity
@Log4j2
@MapperScan(basePackages = {"org.scoula.security.account.mapper"})
@ComponentScan(basePackages = {"org.scoula.security"})
@RequiredArgsConstructor
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    private final UserDetailsService userDetailsService;

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }
}
```

SecurityConfig.java

```
// 문자셋 필터
public CharacterEncodingFilter encodingFilter() {
    CharacterEncodingFilter encodingFilter = new CharacterEncodingFilter();
    encodingFilter.setEncoding("UTF-8");
    encodingFilter.setForceEncoding(true);
    return encodingFilter;
}
```

```
@Override
public void configure(HttpSecurity http) throws Exception {
    http.addFilterBefore(encodingFilter(), CsrfFilter.class);
```

```
// 경로별 접근 권한 설정
http.authorizeRequests()
    .antMatchers("/security/all").permitAll()
    .antMatchers("/security/admin").access("hasRole('ROLE_ADMIN')")
    .antMatchers("/security/member").access("hasRole('ROLE_MEMBER')");
```

webconfig 문자필터와 대응함.
차이점은 이건
security에 적용된 경로에서
동작함

각 경로별로 접근 권한 설정 지정

이 설정에 의해서 401 403 상황 에러 결정남

SecurityConfig.java

폼기반에서는 설정을 통해서 지정하면돼
커스텀할 필요 없음

```
http.formLogin()
    .loginPage("/security/login")
    .loginProcessingUrl("/security/login")
    .defaultSuccessUrl("/");

http.logout()
    // 로그아웃 설정 시작
    .logoutUrl("/security/logout") // POST: 로그아웃 호출 url
    .invalidateHttpSession(true) // 세션 invalidate
    .deleteCookies("remember-me", "JSESSION-ID") // 삭제할 쿠키 목록
    .logoutSuccessUrl("/security/logout"); // GET: 로그아웃 이후 이동할 페이지
}

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    log.info("configure .....");
    auth
        .userService(userDetailsService)
        .passwordEncoder(passwordEncoder());
}
}
```