SecurityConfig 전체 코드

[KB] IT's Your Life

SecurityConfig.java

```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 {

    @Autowired
    private JwtUsernamePasswordAuthenticationFilter jwtUsernamePasswordAuthenticationFilter;

    private final UserDetailsService userDetailsService;
    private final AuthenticationErrorFilter authenticationErrorFilter;
    private final JwtAuthenticationFilter jwtAuthenticationFilter;

    private final CustomAccessDeniedHandler accessDeniedHandler;
    private final CustomAuthenticationEntryPoint authenticationEntryPoint;
```

SecurityConfig 전체 코드

✎ SecurityConfig.java

```java
@Bean // AuthenticationManager 빈 등록
public AuthenticationManager authenticationManager() throws Exception {
    return super.authenticationManager();
}

@Bean  // 비밀번호 인코더 빈 등록
public PasswordEncoder passwordEncoder() {
    return new BCryptPasswordEncoder();
}

// Authentication Manger  구성
@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth
        .userDetailsService(userDetailsService)
        .passwordEncoder(passwordEncoder());
}
```

SecurityConfig.java

```java
@Bean // cross origin 접근 허용
public CorsFilter corsFilter() {
    UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
    CorsConfiguration config = new CorsConfiguration();
    config.setAllowCredentials(true);
    config.addAllowedOriginPattern("*");
    config.addAllowedHeader("*");
    config.addAllowedMethod("*");
    source.registerCorsConfiguration("/**", config);
    return new CorsFilter(source);
}

// 접근 제한 무시 경로 설정 – resource
@Override
public void configure(WebSecurity web) throws Exception {
    web.ignoring().antMatchers("/assets/**", "/*", "/api/member/**");
}
```

SecurityConfig 전체 코드

✎ SecurityConfig.java

```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 {
        // 한글 인코딩 필터 설정
    http.addFilterBefore(encodingFilter(), CsrfFilter.class)
        // 인증 에러 필터
        .addFilterBefore(authenticationErrorFilter, UsernamePasswordAuthenticationFilter.class)
        // Jwt 인증 필터
        .addFilterBefore(jwtAuthenticationFilter, UsernamePasswordAuthenticationFilter.class)
        // 로그인 인증 필터
        .addFilterBefore(jwtUsernamePasswordAuthenticationFilter, UsernamePasswordAuthenticationFilter.class);
```

SecurityConfig.java

```java
// 예외 처리 설정
http
    .exceptionHandling()
    .authenticationEntryPoint(authenticationEntryPoint)
    .accessDeniedHandler(accessDeniedHandler);

http
    .authorizeRequests()
    .antMatchers(HttpMethod.OPTIONS).permitAll()
    .antMatchers("/api/security/all").permitAll()
    .antMatchers("/api/security/member").access("hasRole('ROLE_MEMBER')")
    .antMatchers("/api/security/admin").access("hasRole('ROLE_ADMIN')")
    .anyRequest().authenticated();

http.httpBasic().disable()                      // 기본 HTTP 인증 비활성화
    .csrf().disable()      // CSRF 비활성화
    .formLogin().disable()  // formLogin 비활성화 □ 관련 필터 해제
    .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS); // 세션 생성 모드 설정
    }
}
```