

2025년 상반기 K-디지털 트레이닝

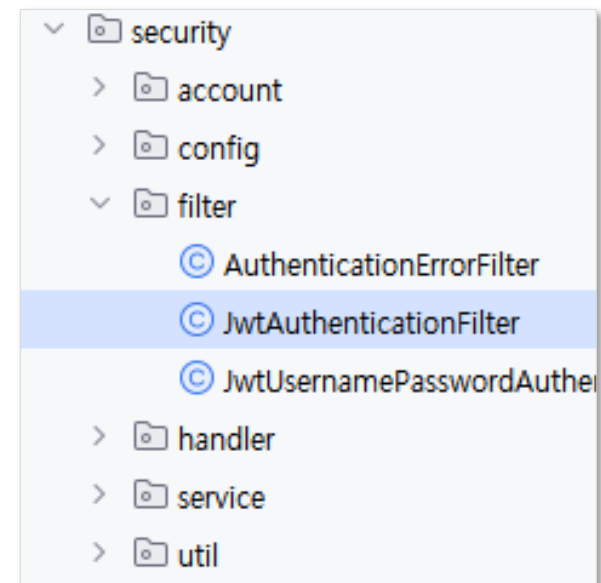
Spring Security JWT 인증2

[KB] IT's Your Life

✓ JwtAuthenticationFilter를 정의하세요.

○ OncePerRequestFilter 상속

- 컨트롤러가 forward 했을 때 필터를 다시 통과하게 됨
→ 이미 보안 필터는 통과했는데, 또 통과하게 됨
- 요청당 한 번만 필터가 동작하도록 해줌



security.filter.JwtAuthenticationFilter.java

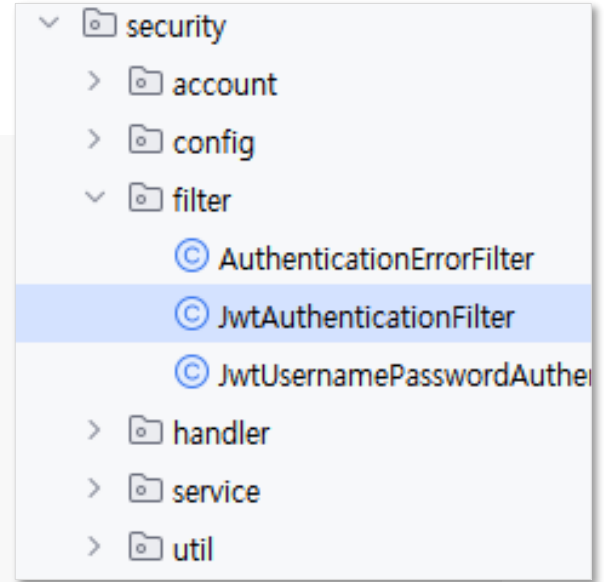
```
package org.scoula.security.filter;

...

@Component
@Log4j2
@RequiredArgsConstructor
public class JwtAuthenticationFilter extends OncePerRequestFilter {
    public static final String AUTHORIZATION_HEADER = "Authorization";
    public static final String BEARER_PREFIX = "Bearer "; // 끝에 공백 있음

    private final JwtProcessor jwtProcessor;
    private final UserDetailsService userDetailsService;

    private Authentication getAuthentication(String token) {
        String username = jwtProcessor.getUsername(token);
        UserDetails principi = userDetailsService.loadUserByUsername(username);
        return new UsernamePasswordAuthenticationToken(principi, null, principi.getAuthorities());
    }
}
```



security.filter.JwtAuthenticationFilter.java

@Override

```
protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain)
    throws ServletException, IOException {
```

```
    String bearerToken = request.getHeader(AUTHORIZATION_HEADER);
```

```
    if (bearerToken != null && bearerToken.startsWith(BEARER_PREFIX)) {
        String token = bearerToken.substring(BEARER_PREFIX.length());
```

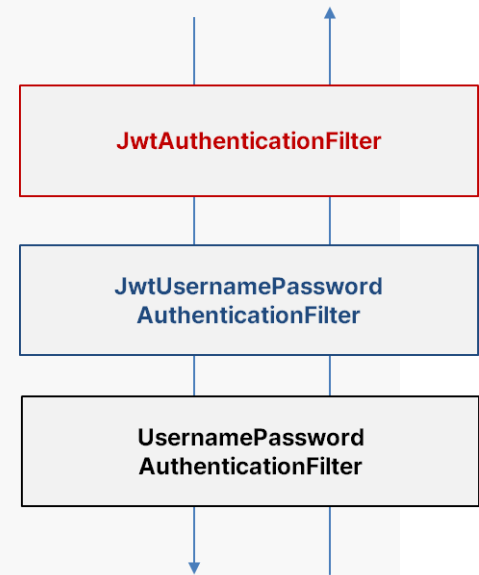
```
        // 토큰에서 사용자 정보 추출 및 Authentication 객체 구성 후 SecurityContext에 저장
        Authentication authentication = getAuthentication(token);
        SecurityContextHolder.getContext().setAuthentication(authentication);
    }
```

```
    super.doFilter(request, response, filterChain);
}
```

- ✓ JwtAuthenticationFilter를 SecurityConfig에 등록하세요.

security.config.SecurityConfig.java

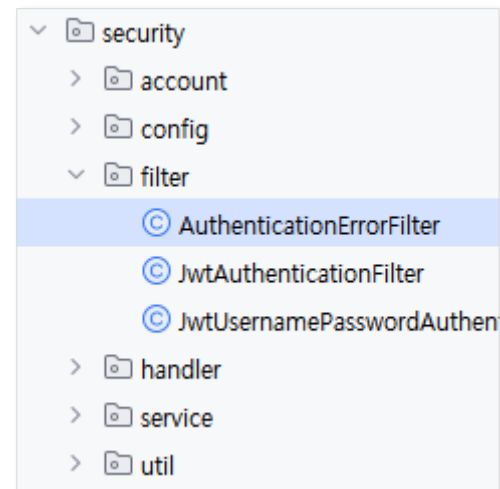
```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
    private final UserDetailsService userDetailsService;  
  
    private final JwtAuthenticationFilter jwtAuthenticationFilter;  
  
    @Autowired  
    private JwtUsernamePasswordAuthenticationFilter jwtUsernamePasswordAuthenticationFilter;  
  
    ...  
    @Override  
    public void configure(HttpSecurity http) throws Exception {  
        // 한글 인코딩 필터 설정  
        http.addFilterBefore(encodingFilter(), CsrfFilter.class)  
        // Jwt 인증 필터  
        .addFilterBefore(jwtAuthenticationFilter, UsernamePasswordAuthenticationFilter.class)  
        // 로그인 인증 필터  
        .addFilterBefore(jwtUsernamePasswordAuthenticationFilter, UsernamePasswordAuthenticationFilter.class);  
  
        ...  
    }  
}
```



✓ 인증 예외 처리를 하는 AuthenticationErrorFilter를 정의하세요.

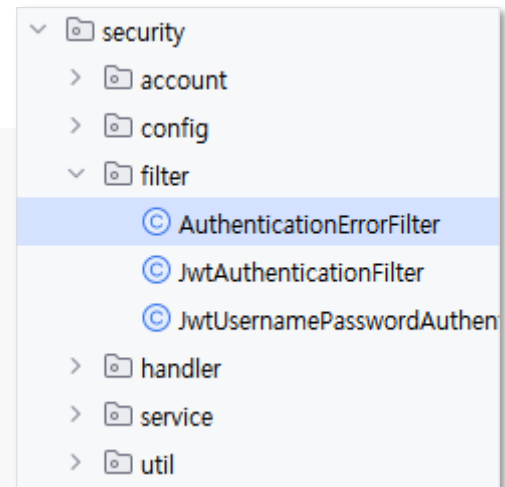
○ 토큰 처리시 예외 발생 가능

- 여러 예외 중 토큰 만기 예외인 경우, 401 인증 에러로 리턴
- 그 외의 예외는 일반 예외 메시지로 처리



security.filter.AuthenticationErrorFilter.java

```
package org.scoula.security.filter;
...
import io.jsonwebtoken.security.SignatureException;
...
@Component
public class AuthenticationErrorFilter extends OncePerRequestFilter {
    @Override
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain)
        throws ServletException, IOException {
        try {
            super.doFilter(request, response, filterChain);
        } catch (ExpiredJwtException e) {
            JsonResponse.sendError(response, HttpStatus.UNAUTHORIZED, "토큰의 유효시간이 지났습니다.");
        } catch (UnsupportedJwtException | MalformedJwtException | SignatureException e) {
            JsonResponse.sendError(response, HttpStatus.UNAUTHORIZED, e.getMessage());
        } catch (ServletException e) {
            JsonResponse.sendError(response, HttpStatus.INTERNAL_SERVER_ERROR, e.getMessage());
        }
    }
}
```



- ✓ **AuthenticationErrorFilter를 SecurityConfig에 등록하세요.**

security.config.SecurityConfig.java

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
    private final UserDetailsService userDetailsService;  
    private final JwtAuthenticationFilter jwtAuthenticationFilter;  
    private final AuthenticationErrorFilter authenticationErrorFilter;  
    @Autowired  
    private JwtUsernamePasswordAuthenticationFilter jwtUsernamePasswordAuthenticationFilter;  
    ...  
    @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);  
        ...  
    }  
}
```

