

API기반!!!



JWT 기반!!

2025년 상반기 K-디지털 트레이닝



















SecurityConfig 전체 코드

[KB] IT's Your Life

SecurityConfig 전체 코드

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 {
```

@Autowired                      

SecurityConfig 전체 코드

SecurityConfig.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());
}
```

=> config 생성 이후에 준비되는 bean

필터 몇가지와 핸들러 몇가지 준비하면 돼

SecurityConfig.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/**");
}
```

이 부분이
뷰 연동 파트
정적파트

index.html
single page
가
들어가는 곳


SecurityConfig 전체 코드

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 {
    // 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);
}
```

필터순서 지정



SecurityConfig 전체 코드

SecurityConfig.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()
.csrf().disable()
.formLogin().disable()
.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);

}
```

예외 지정

접근 지정

// 기본 HTTP 인증 비활성화

// CSRF 비활성화

// formLogin 비활성화 □ 관련 필터 해제

// 세션 생성 모드 설정

비활성화 해버려
필요없는 것들

스프링부트는 구조는 안바뀌는데 설정하는게 엄청 쉬워진.
동작 설정과 원리는 그대로.