

## Spring Boot – Spring Boot 2.7.x to Spring boot 3.x.x migrations

antMatchers() -> make very slowly

### Preparing for Spring Boot 3.0

<https://spring.io/blog/2022/05/24/preparing-for-spring-boot-3-0>

In pom.xml: - use version 3 -> change it to 3.0.0

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.7.5</version>
  <relativePath /> <!-- lookup parent from repository -->
</parent>
```

Change it to 6

```
<dependency>
  <groupId>org.thymeleaf.extras</groupId>
  <artifactId>thymeleaf-extras-springsecurity5</artifactId>
</dependency>
```

Change it to 3.0.6

```
<dependency>
  <groupId>org.thymeleaf.extras</groupId>
  <artifactId>thymeleaf-extras-java8time</artifactId>
  <version>3.0.4.RELEASE</version>
</dependency>
```

Source Action -> Organize Imports

Use Jakarta for imports if there are any error causing. (For me it was causing in models)

@EnableGlobalMethodSecurity is deprecated ; so change to @EnableMethodSecurity

```
@EnableMethodSecurity(prePostEnabled = true, securedEnabled = true, jsr250Enabled = true)
```

Can remove the parameters

## WebSecurityConfig.java

```
package org.studyeasy.SpringBlog.security;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.Customizer;
import org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
@EnableWebSecurity
@EnableMethodSecurity
public class WebSecurityConfig {
    private static final String[] WHITELIST = {
        "/",
        "/login",
        "/register",
        "/db-console/**",
        "/resources/**",
        "/posts/**"
    };

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

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .headers(headers -> headers.frameOptions().sameOrigin())
            .authorizeHttpRequests(requests -> requests
                .requestMatchers(WHITELIST).permitAll()
                .requestMatchers("/profile/**").authenticated()
                .requestMatchers("/update_photo/**").authenticated())
    }
}
```

```
.requestMatchers("/posts/add/**").authenticated()

.requestMatchers("/admin/**").hasRole("ADMIN")
.requestMatchers("/editor/**").hasAnyRole("ADMIN", "EDITOR"))
.formLogin(login -> login
    .loginPage("/login").loginProcessingUrl("/login")
    .usernameParameter("email").passwordParameter("password")
    .defaultSuccessUrl("/", true).failureUrl("/login?error")
    .permitAll())
.logout(logout -> logout.logoutUrl("/logout")
    .logoutSuccessUrl("/"))
    .rememberMe(me -> me.rememberMeParameter("remember-me"))
.httpBasic(Customizer.withDefaults());

http.csrf(csrf -> csrf.disable());
http.headers(headers -> headers.frameOptions().disable());

return http.build();

}

}
```