



WEB ILOVALARDA SPRING SECURITY ORQALI

AUTENTIFIKATSIYA VA AVTORIZATSIYA JARAYONINI AMALGA
OSHIRISH*Qahorova Sarvinoz Fazliddin qizi*NavDKTU 15-23 MSM guruh
talabasi*Abjalilov Ozodjon Shomurat o'g'li*

NavDKTU MSS kafedrasи assistenti

Spring Security bu Spring frameworkning asosiy modullaridan biri bo'lib, u loyihalarni himoyalashning bir nechta usullarini hamda ko'rinishlarini taqdim etadi. Spring Security framework autentifikatsiya (foydalanuvchini tekshirish) va avtorizatsiya (foydalanuvchining tizimda qanday huquqlari borligini tekshirish) jarayonlarini osonlashtiradi. Spring Securityda murojat qilayotgan foydalanuvchini tekshirish va shu foydalanuvchining murojat qilinayotgan API ruxsati bor yoki yo'qligini yozish kerak bo'ladi. Shu yozilgan konfiguratsiyalardan kelib chiqib, Spring Security qaysi API-ni qaysi foydalanuvchilar ishlata olishi mumkin ekanligini biladi. Spring Security asosan veb-ilovalarni himoya qilish uchun keng qo'llaniladi. Spring Securityning yengilligi shundaki, uni talabdan kelib chiqib dasturchi tamondan xohlagancha o'zgartirish mumkun. Spring Security frameworki dasturchilarga xavfsizlikni ta'minlashda katta yordam beradi, chunki u kengaytiriladigan va moslashuvchan framework hisoblanadi. U oddiy autentifikatsiyadan tortib, murakkab xavfsizlik talablariga qadar keng qamrovli yechimlarni taqdim etadi [1-2].

Autentifikatsiya (Authentication). Foydalanuvchining identifikatsiya qilish. Ya'ni murojat qilinayotgan foydalanuvchining login va paroli to'g'ri yoki xato ekanligini tekshirish, uning bloklanmaganligini tekshirish vazifasini bajaradi.

Avtorizatsiya. Foydalanuvchining tizimdagи huquqlarini tekshirish, Ya'ni foydalanuvchi murojat qilayotgan API larni ishlatishga ruxsati bor yoki yo'qligini tekshirish hisoblanadi. Rollarga asoslangan xavfsizlik (RBAC - Role-Based Access



Control) va ruxsatlarga asoslangan xavfsizlik (PBAC - Permission-Based Access Control) [3-4].

Himoya mexanizmlari. CSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), va boshqa hujumlarga qarshi himoya uchun ishlataladi. Ya’ni CSS, CSRF va shu kabi hujumlardan himoyalanish usullari mavjud. HTTPS orqali ulanishni ishlatish mumkun. Shuningdek SSL sertifikatlarini qo‘llab-quvvatlash [5].

```
import java.util.List;

@Configuration
@EnableWebSecurity
public class SpringConfig {

    @Bean
    public AuthenticationProvider authenticationProvider() {
        String password = "qFhesswrd_7777";
        System.out.println("Using generated security password :->" + password);
        UserDetailsService userDetailsService = new InMemoryUserDetailsManager(user);
        return authenticationProvider;
    }

    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .authorizeHttpRequests((authz) -> authz
                .anyRequest().authenticated())
            .formLogin((form) -> form
                .loginPage("/login")
                .username("username")
                .password("new123")
                .permitAll())
            .logout((logout) -> logout
                .logoutUrl("/logout")
                .deleteCookies("JSESSIONID"))
            .exceptionHandling((exception) -> exception
                .accessDeniedPage("/access-denied"));
    }
}
```

1-rasm. Foydalanuvchining identifikatsiya qilish dasturi

```
public class SpringConfig {
    @Bean
    public SecurityFilterChain security(UrlBasedHttpSecurity http) throws Exception {
        // enable csrf protection (disabled by default)
        // http.csrf(AbstractHttpConfigurer::disable);
        http.authorizeHttpRequests(AuthorizationRequestMatcher::isAuthenticated, authenticationManagerRequestMatcherRegistry -> {
            authenticationManagerRequestMatcherRegistry
                .requestMatchers(HttpServletRequestMethod.GET, "/*").permitAll()
                .requestMatchers(HttpServletRequestMethod.POST, "/*").permitAll()
                .requestMatchers("/path/to/resource/1/*").hasRole("USER")
                .requestMatchers(HttpServletRequestMethod.PUT, "/*").permitAll()
                .anyRequest().authenticated();
        });
        http.httpBasic(Customizer.withDefaults());
        http.cors(AbstractHttpConfigurer::enable);
        http.cors(AbstractHttpConfigurer::enable);
        return http.build();
    }
}
```

2-rasm. Foydalanuvchining tizimdagi huquqlarini tekshirish dasturi

Integratsiya. Spring Boot bilan oson integratsiya qilinadi. LDAP, JDBC, va boshqa ma'lumotlar bazalari bilan ishslash imkoniyatlari mavjud.

Spring Security ishslash tartibi. Spring Security loyihaga xavfsizlik filtrlari zanjiri ko'rinishida qo'shiladi. Ushbu filtrlardan har biri ma'lum bir xavfsizlik funksiyasini bajaradi. Masalan, autentifikatsiya, avtorizatsiya, sessiya boshqaruvi va boshqalar kiradi. Foydalanuvchi murojat qilganda, uning murojatini ushbu filtrlardan o'tkaziladi va har bir filtr o'zining vazifasini bajaradi [6]. Agar qaysidir filtr zanjiridan o'tmasa uning murojati ortga qaytarilib yuboriladi. Barchasidan muvaffaqiyatli o'tsa, murojat qilayotgan APIni ishlatish mumkin.

Xulosa. Spring Security foydalanuvchini aniqlash, ruxsatlarni tekshirish, sessiyalarni boshqarish, parolni shifrlash, CSRF, CORS, va boshqalarni qo'llab-quvvatlaydi. Spring Boot bilan mukammal integratsiyalashgan bo'lib, spring-boot-starter-security orqali tez va oson sozlash mumkin. Spring Securityni o'zehtiyojlarga qarab moslashtirish juda oson. Spring Security orqali foydalanuvchilarga rollar (ROLE_ADMIN, ROLE_USER va hokazo) berish va shunga asoslangan resurslarga kirishni boshqarish oson. Spring Security JWT (JSON Web Token) bilan xavfsiz REST API yaratishda keng qo'llaniladi. OAuth2 va OpenID Connect orqali tashqi tizimlar (Google, Facebook login) bilan autentifikatsiyani amalga oshirish mumkin. Spring Security keng qo'llaniladi, katta jamoa tomonidan qo'llab-quvvatlanadi va rasmiy hujjatlari boy sanaladi. Bu esa o'rganishni va muammolarni yechishni osonlashtiradi. Spring Security muntazam ravishda yangilanadi va zamonaviy xavfsizlik tahdidlariga moslab takomillashtiriladi. Bular esa uning asosiy afzalliklaridan sanaladi.

Foydalanilgan adabiyotlar:

1. Madden N. "Api Securitiy in Action" o'quv qo'llanma 2020. -576 bet.
2. Walls C. "Spring in Action, Sixth Edition" o'quv qo'llanma 2020. -520 bet.
3. Spilca L. "Spring Start Here" o'quv qo'llanma 2021.-416 bet.
4. Carnell J. "Spring Microservices in Action, Second Edition" o'quv qo'llanma 2021.-448 bet.

5. Юсупбеков Н. Р. и др. Ноаниқ мантиқ асосида интеллектуал бошқариш тизимларини ишлаб чиқиши //Journal of Advances in Engineering Technology. – 2020. – №. 2. – С. 20-25.

6. Махмудов Г. Б., Сайдова А. Х., Мохилова Н. Т. Моделирование нечеткой логики для управления процессом бактериального окисления концентратов в реакторах с мешалкой //Современные инновации, системы и технологии. – 2022. – Т. 2. – №. 2. – С. 0201-0214.