

Spring Security

- Rozbudowane, elastyczne i konfigurowalne środowisko uwierzytelniania oraz kontroli dostępu
- Jest to de facto standardem zabezpieczenia aplikacji opartych o Spring framework (warstwa webowa, komponenty)
- Umożliwia integrację z wieloma popularnymi rozwiązaniami m.in. LDAP, Open ID, JAAS, CAS
- Konfiguracja odbywa się z użyciem XML, Java Configuration lub adnotacji

@EnableWebSecurity

@Configuration

```
public class SecSecurityConfig extends WebSecurityConfigurerAdapter {
```

```
    @Autowired
```

```
    private PasswordEncoder passwordEncoder;
```

```
    @Bean
```

```
    public PasswordEncoder passwordEncoder() {
```

```
        return new BCryptPasswordEncoder();
```

```
    }
```

```
    @Override
```

```
    protected void configure(final AuthenticationManagerBuilder auth)
```

```
        auth.inMemoryAuthentication()
```

```
            .withUser("client").password(passwordEncoder.encode("1234567890"))
```

```
            .withUser("employee").password(passwordEncoder.encode("1234567890"))
```

```
    }
```

```
    @Override
```

```
    protected void configure(final HttpSecurity http) throws Exception
```

```
        http.authorizeRequests()
```

```
            .antMatchers("/management/**").hasRole("admin")
```

```
            .antMatchers("/public*").permitAll()
```

```
            .and()
```

```
            .formLogin()
```

```
                .loginPage("/login.html")
```

```
                .defaultSuccessUrl("/home.html")
```

```
            .and()
```

```
            .logout()
```

```
                .logoutUrl("/perform_logout")
```

```
                .deleteCookies("JSESSIONID")
```

```
                .logoutSuccessHandler(logoutSuccessHandler());
```

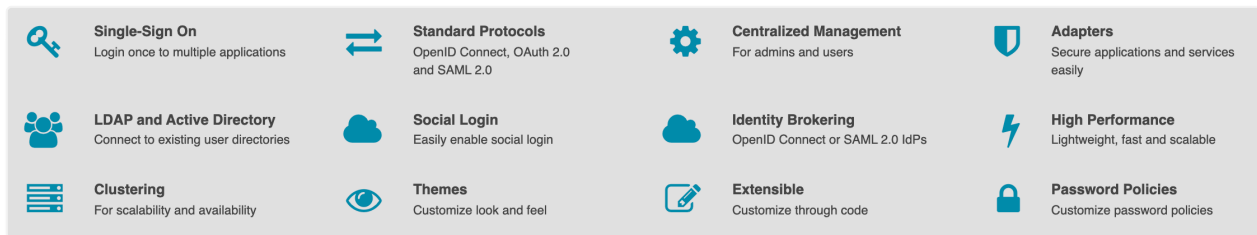
```
    }
```

```
}
```

<https://spring.io/projects/spring-security>

Keycloak

- Otwarte rozwiązanie umożliwiające kontrolę dostępu i zarządzanie tożsamością użytkowników
- Niezależne od platformy i języka programowania
- Oparte o standardy: **OpenID Connect**, **SAML 2**, **Kerberos**
- Łatwe w integracji
- Udostępnia webową konsolę administracyjną oraz REST API
- Alternatywne rozwiązania: Auth0, Okta, Gluu, OpenAM



<https://www.keycloak.org>

OAuth2

- Otwarty protokół **autoryzacji** dostępu, wykorzystywany m.in. przez Amazon, Google, Facebook, Microsoft oraz Twitter
- Pozwala udostępniać aplikacjom i stronom trzecim informacje przechowywane u innych dostawców usług
- Terminologia:
 - *Resource owner* – osoba lub podmiot będący właścicielem chronionego zasobu
 - *Client* – aplikacja, która chce uzyskać dostęp do zasobu
 - *Authorization server* – serwer udzielający poświadczenia wymaganego do uzyskania dostępu chronionego zasobu
 - *Resource server* – serwer zawierający chronione zasoby
 - *Scope* – zbiór uprawnień (namespaces) chronionych zasobów
 - *Access token* – token wystawiany przez serwer autoryzacyjny, wykorzystywany aby uzyskać dostęp do zasobów

<https://oauth.net>

<https://www.manning.com/books/oauth-2-in-action>

<https://sekurak.pl/oauth-2-0-jak-dziala-jak-testowac-problemy-bezpieczenstwa>

<https://medium.com/@darutk/diagrams-and-movies-of-all-the-oauth-2-0-flows-194f3c3ade85>

<https://developer.okta.com/blog/2019/10/21/illustrated-guide-to-oauth-and-oidc>

OpenID Connect

- Warstwa uwierzytelniająca oparta na protokole OAuth 2.0
- Standard kontrolowany przez OpenID Foundation
- Umożliwia klientom weryfikację tożsamości użytkownika na podstawie uwierzytelnienia przeprowadzonego przez serwer autoryzacji, a także uzyskanie podstawowych informacji o profilu użytkownika
- Oferuje scenariusze, w których jedno logowanie może być używane w wielu aplikacjach (ang. Single Sign-On)

<https://openid.net>

Demo

- Uruchamianie serwera Keycloak (standalone, Docker)
- Konfiguracja: realm, clients, users, roles, identity providers, 2FA, themes
- Integracja: Spring Boot, Spring Security

<https://github.com/landrzejewski/webinar-keycloak>

Spring framework masterclass

- Kompleksowe szkolenie w formie materiału video oraz ćwiczeń praktycznych
- Kurs podzielony jest na 12 modułów obejmujących kluczowe zagadnienia z zakresu użycia frameworka
- Dlaczego warto wybrać ten kurs:
 - Opanujesz Spring framework w stopniu pozwalającym na jego wykorzystanie w realnym projekcie
 - Podniesiesz swoje umiejętności, a także wartość na rynku dzięki czemu łatwiej otrzymasz nową pracę, awans lub podwyżkę
 - Otrzymasz wsparcie doświadczonego mentora oraz społeczności związanej z kursem
 - Unikniesz błędów popełnianych przez osoby zaczynające samodzielną naukę
 - Systematyczna aktualizacje materiałów i praca z najnowszą wersją Spring
 - Warsztatowy charakter zajęć - wspólna realizacja projektu i ćwiczenia do samodzielnego rozwiązania
 - Dostęp do regularnych konsultacji on-line z mentorem oraz innymi uczestnikami

<https://kursy.sages.pl/spring-masterclass>