Szkolenie: Capstone Courseware
117D Spring Security

**FORMA SZKOLENIA** | **MATERIAŁY SZKOLENIOWE** | **CENA** | **CZAS TRWANIA**
--- | --- | --- | ---
Stacjonarne | Tradycyjne | 3750 PLN NETTO* | 4 dni
Stacjonarne | Tablet CTAB | 4150 PLN NETTO* | 4 dni
Metoda dlearning | Tradycyjne | 3750 PLN NETTO* | 4 dni
Metoda dlearning | Tablet CTAB | 3750 PLN NETTO* | 4 dni

* (+VAT zgodnie z obowiązującą stawką w dniu wystawienia faktury)

**LOKALIZACJE**

Kraków - ul. Tatarska 5, II piętro, godz. 9:00 - 16:00
Warszawa - ul. Bielska 17, godz. 9:00 - 16:00

**Cel szkolenia:**

Version 3.2

This in-depth course introduces the Java web developer to the **Spring Security framework**. We start with an overview and practical exercises in basic usage: XML configuration for authentication and URL-based authorization. Then we start to dig into Spring Security as a Java model, and develop advanced techniques including custom user realms, custom authorization constraints, method-based authorization, and instance-based authorization.

We then explore two increasingly popular extensions to Spring Security. We consider the Security Assertions Markup Language, or SAML, and the wide range of identity and security features it offers -- but quickly focus on it’s support for single sign-on (SSO), and learn how the Spring Security SAML Extension enables applications to interact with SAML identity providers to implement SSO and single logout. And we look at OAuth for Spring Security, which enables third-party authorization scenarios, and learn how to implement both the server and client sides of the OAuth 2.0 flow.

Note that the course does not give much background on general web-application security -- pros and cons of HTTP BASIC, DIGEST, and form-based authentication strategies, cross-site scripting, Injection, CSRF, etc. Course 121, "Securing Java Web Applications," makes a nice compliment to Course 117D, and custom hybrids of the two courses are available.

**Learning Objectives**

- Configure Spring Security for HTTP BASIC authentication.
- Implement form-based authentication.
- Configure other authentication features including remember-me, anonymous users, and logout.
- Apply authorization constraints to URLs and URL patterns.
- Bind authorization roles to user accounts in relational databases.
- Plug application-specific user realms into Spring Security by implementing UserDetailsService.
- Implement application-specific authorization constraints as AccessDecisionVoters.
- Fix authorization constraints over individual methods of service beans, in lieu of URL authorization or in tandem with it.
- Express user identity in terms of SAMLs.
- Implement SAML SSO from the service-provider side.
- Implement OAuth 2.0 authorization-server and resource-server roles.
- Implement an OAuth 2.0 client.

Plan szkolenia:

- **Spring Security**
  - Acquiring and Integrating Spring Security
  - Relationship to Spring
  - Relationship to Java EE Standards
  - Basic Configuration
  - How It Works
  - Integration: LDAP, CAS, X.509, OpenID, etc.
  - Integration: JAAS

- **Authentication**
  - TheConfiguration
  - TheConstraint
  - Login Form Design
  - "Remember Me"
  - Anonymous "Authentication"
  - Logout
  - The JDBC Authentication Provider
  - The Authentication/Authorization Schema
  - Using Hashed Passwords
  - Why Hashing Isn't Enough
  - Using Salts
  - PasswordEncoder and SaltSource
  - Key Lengthening
  - Channel Security
Session Management

URL Authorization
  - URL Authorization
  - Programmatic Authorization: Servlets
  - Programmatic Authorization: Spring Security
  - Role-Based Presentation
  - The Spring Security Tag Library

Under the Hood: Authentication
  - The Spring Security API
  - The Filter Chain
  - Authentication Manager and Providers
  - The Security Context
  - Plug-In Points
  - Implementing UserDetailsService
  - Connecting User Details to the Domain Model

Under the Hood: Authorization
  - Authorization
  - FilterSecurityInterceptor and Friends
  - The AccessDecisionManager
  - Voting
  - Configuration Attributes
  - Access-Decision Strategies
  - Implementing AccessDecisionVoter
  - The Role Prefix

Method and Instance Authorization
  - Method Authorization
  - Using Spring AOP
  - XML vs. Annotations
  - @PreAuthorize and @PostAuthorize
  - Spring EL for Authorization
  - @PreFilter and @PostFilter
  - Domain-Object Authorization
  - The ACL Schema
  - Interface Model
  - ACL-Based Presentation

Introduction to SAML
History of SAML
Assertions
Protocol
Bindings
Profiles
Using OpenSAML

SAML Assertions and Protocol
"Vouching for" a User
Assertions and Subjects
NameID Types
Authentication Contexts
Requests, Queries, and Responses
Attribute Queries
SAML and XML Signature

SAML Bindings
Speaking "Through" the Browser
The SOAP Binding
SAML Over HTTP
The Redirect, POST, and Artifact Bindings
The PAOS Binding
The URI Binding

Federated Identity and SSO
SAML 2.0 Federations
Single Sign-On
Account Linking and Persistent Pseudonyms
Transient Pseudonyms
Name ID Mapping
Single Logout
Federation Termination

The Spring Security SAML Extension
The Spring Security SAML Extension
The SAML Entry Point
The SAML Filter Chain
The SSO Processing Filters
IdP Discovery
Login and Logout Handlers
- Configuring OpenAM
- Configuring an SP
- Customization
- Combining SSO and Other Authentication Styles
- Authorization and Attributes
  - OAuth for Spring Security
    - Third-Party Authorization
    - OAuth
    - Roles and Initial Flow
    - Grant Types
    - Access Tokens
    - The Google OAuth API
    - OAuth for Spring Security
    - Client-Details Services
    - Token Services
    - The AuthorizationEndpoint
    - The TokenEndpoint
    - The UserApprovalHandler
    - The Resource-Server Filter
    - The ScopeVoter
    - The OAuth-Aware RestTemplate
    - AccessTokenProviders
    - The OAuth Redirecting Filter

**Wymagania:**

- **Java programming:** Course 103 is excellent preparation.
- Experience with the Spring framework -- consider at Course 117 on Core Spring as a minimum, or Course 117A for an understanding of Spring MVC.
- **Basic knowledge of XML:** Course 501.
- Some servlets and/or JSP experience will be beneficial for purposes of understanding the impact of each security feature that we configure. There is no web-application coding involved in the course.

**Poziom trudności**
Certyfikaty:

The participants will obtain certificates signed by Capstone Courseware.

Prowadzący:

Authorized Capstone Courseware Trainer.