Spring Security

Spring Security is a framework that provides authentication, authorization, and protection against common attacks.it is the de-facto standard for securing Spring-based applications.

Agenda

- What is Spring security?
- Setting up Spring Security
- How Basic Spring Security Flow Works
- Simple customizations of spring security
  - Login Page
  - Custom Log in Controller
- Global Method Security

Overview

- Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications.
- Spring Security is a framework that focuses on providing both authentication and authorization to Java applications.
- Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements

Features

- Comprehensive and extensible support for both Authentication and Authorization
- Protection against attacks like
  - Session fixation / hijacking
  - Click jacking
  - Cross site request forgery, etc.
- Servlet API integration
- Optional integration with Spring Web MVC
- Much more…

Spring Security

- Formerly Known as Acegi Security (2003) Ben Alex
  - Part of Spring (2007)
- **Authentication**
  - Database
  - LDAP
  - CAS (**Central Authentication Service)**
  - OpenID
  - Pre-Authentication,
  - Custom etc.
- **Authorization**
  - URL based
  - Method Based (AOP)
- Extensions

Authentication

- "Authentication" is the process of establishing a principal is who they claim to be (a "principal" generally means a user, device or some other system which can perform an action in your application)
- Authentication is the act of confirming the truth of an attribute of a single piece of data claimed true by an entity

- Authentication Factors
  - Something you know
    - password, PIN
  - Something You have
    - Smart card, Token
  - Something you are
    - Fingerprint

Authentication

- Spring Security currently supports authentication integration with all of these technologies:
  - HTTP BASIC authentication headers
  - HTTP Digest authentication headers
  - HTTP X.509 client certificate exchange
  - LDAP
  - Form-based authentication
  - OpenID authentication
  - Jasig Central Authentication Service
  - Kerberos
  - SAML
  - OAuth2

Authorization

- "Authorization" refers to the process of deciding whether a principal is allowed to perform an action within your application.
- Authorization is the function of specifying access rights/privileges to resources,
- The process of ensuring only authorized right are exercised
- The Process of determining rights

- How do users Receive rights?
  - Access Control Lists (ACLs)
    - Discretionary Access control (DAC)
    - Role Based Access Control (RBAC)
    - Mandatory Access Contorl (MAC)

Authorization

- Spring Security provides a deep set of authorization capabilities.

- There are three main areas of interest:
  - Authorizing web requests (URL Based)
  - Authorizing whether methods can be invoked (AOP)
  - Authorizing access to individual domain object instances

Intro

- What is Spring Security and advantage
- Introducing the Latest version
- Getting Spring security core modules and runtime environment
- Core Components
- Authentication basics
- Authentication in a Web Application

Spring Security

- Spring Security Overview
- Spring Security advantages
- A simple Spring MVC application

Spring Security

- Security framework for Java based application
- Utilizes the power of Spring - Dependency Injection
- Power of Authentication and Authorization
- Wide range of Integration support with different technologies such as LDAP, Single sign-on , Kerberos and so on

Getting Spring Security

- Spring security modules
- Configure pom.xml(Maven) to fetch spring-security libraries
- Runtime environment

Spring Boot Support

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

Without Boot

```
<dependencies>
  <!-- ... other dependency elements ... -->
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
  </dependency>
</dependencies>
```

Core Components

- Authentication
  - UsernamePasswordAuthenticationToken
- AuthenticationManager
  - ProviderManager
    - AuthenticationProvider
      - Supports(), authenticate()
- GrantedAuthority
- SecurityContextHolder
- SecurityContext

Core Components

- Principal - Object principal
  - SecurityContextHolder.getContext().getAuthentication().getPrincipal()
- UserDetails
  - principal instanceof UserDetails
- UserDetailsService –
  - UserDetails loadUserByUserName(String username) thorw UserNotFoundException

Authentication Sample

- Spring Security Internal Demo