

=> NOTE : We can achieve security in J2EE technologies i.e. Servlets & JSP, but it is difficult to implement security and was also limited. We can integrate Spring Security with J2EE technologies

I

=> Spring Security :-

-> History :-

- = 2003 : Ben Alex started one project "Acegi Security System for Spring" (not Spring Security)
- = 2004 : Spring Security (Version 1.0) was publicly released under Apache Licence (keeping the "Acegi" name during its initial release)
- = 2008 : Spring Security (Version 2.0) was released, marking the official adoption of the "Spring Security" name for the framework
- = After this new releases were regularly introduced

-> Advantages :-

1. Open Source Lightweight Framework :-
 - = It is open source and lightweight which makes it easy to use
 2. Highly configurable and pluggable :-
 - = Supports Java, Annotations and XML configurations and it can be integrated with various technologies (Spring Data, Spring WebFlux, LDAP, Spring MVC, Spring Boot etc)
 3. Strong security measures :-
 - = Protects against common web threats i.e. session fixation, clickjacking, XSS, CSRF etc
 4. Support modern authentication methods :-
 - = Supports social logins, token-based authentication, multi-factor authentication etc
 5. Extensive community and ecosystem :-
 - = Provides rich documentation, tutorials and libraries
 6. Continuous updates and improvements :-
 - = Regular releases ensure bug fixes, security enhancements and new features
- etc

etc

-> Types of Spring Security (Spring Security Features) :-

= Authentication:-

1. In-memory Authentication:

I - Simple for testing and demos, not recommended for production due to security risks

2. Web-based Authentication:

- HTTP Basic: Prompts security window popup for username and password

- HTTP Digest: Sends credentials securely through hases

3. Form-based Authentication:

- Custom Forms: Username and password login through custom login forms

- Captcha: Captcha integration for enhanced security

4. Database Authentication:-

- JDBC: Authenticate users based on credentials stored in relational database

- Custom Database: Implement custom queries to validate the user credentials

- LDAP (Lightweight Directory Access Protocol): Centralize user management using an LDAP server

5. Social Login:

- OpenID Connect: Single sing-on (SSO) with popular providers i.e. Google or Facebook

- OAuth 2.0: Authenticate users using third-party authorized servers

6. Remeber-me Authentication:-

- Stores a cookie for automatic login after successful authentication

7. Token-based authentication:-

- JWT (JSON Web Tokens): Securely transmit the user information between applications

etc

= Authorization:

1. Role-based Access Control (RBAC)

2. Permission-based Access Control (PBAC)

3. URL-based Access Control

4. API Security

= Additional Features :-

1. CSRF (Cross-Site Request Forgery) Protection
2. XSS (Cross-Site Scripting) Protection

-> NOTE : Spring Security mostly focus on 2 areas i.e. Authentication and Authorization

= Authentication :-

- It's the process of verifying a user's identity to confirm they are who they claim to be
- It's the first level of security before granting access to resources
- There are various types of authentication :-

1. Knowledge Based Authentication :-

- >> Username and password
- >> PIN Codes
- >> Security questions

2. Possession Based Authentication :-

- >> Phone or text messages
- >> Key cards and badges
- >> Access token devices

3. Biometric Authentication :-

- >> Fingerprint
- >> Iris Scan

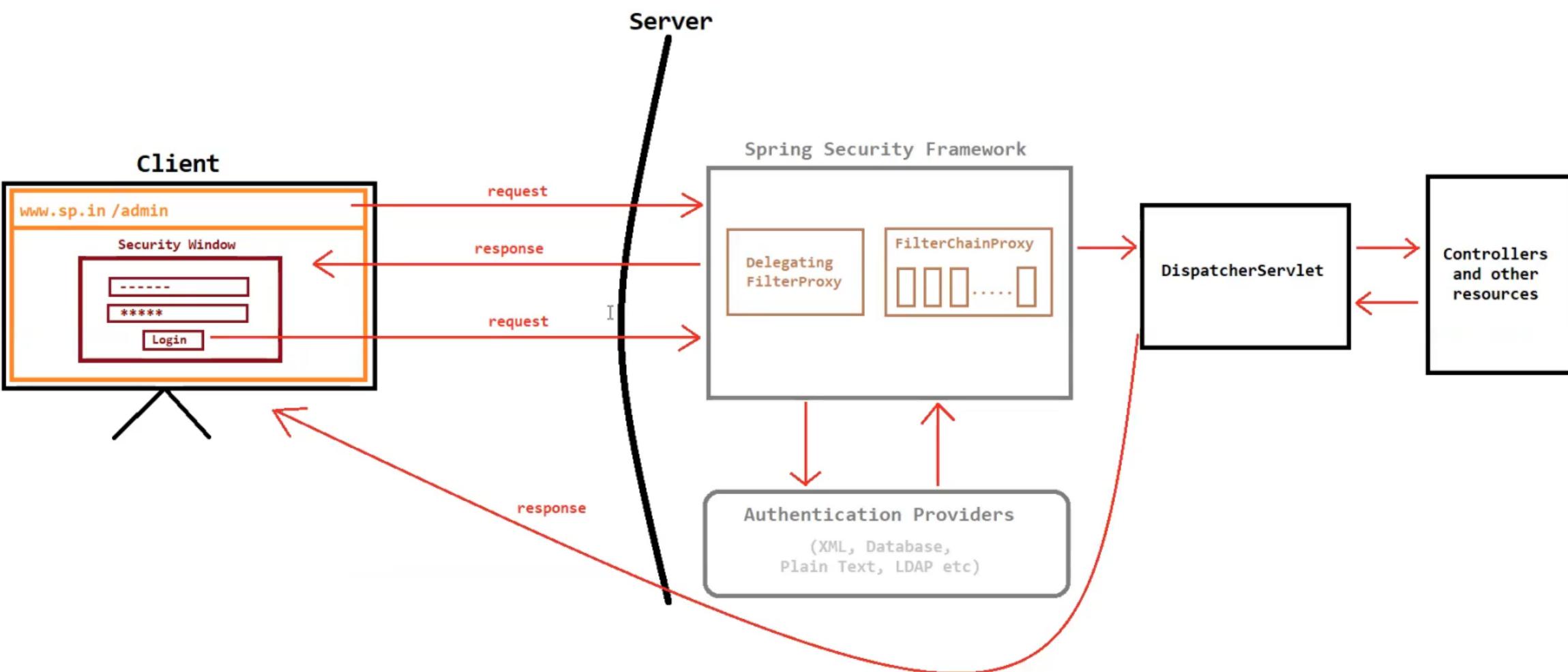
- = Authorization :-
 - It is the process of providing special permission or authority or action based on their roles (admin, user, guest)
 - It comes after successful authentication, granting specific privileges within the application
 - Examples of authorization :-
 1. Role-based authorization : Users assigned specific roles (admin, user, guest) with defined permission
 2. Attribute-based authorization : User attributes (department level) determines access to resources

-> Flow of Spring Security :-

- = Diagram

-> There are 2 important components of Spring Security Framework :-

- 1. DelegatingFilterProxy:
 - It is a class which acts as a central entry point for Spring Security
 - It intercepts all the requests and delegates them to the right security filter based on the rules
 - It makes security setup simple and flexible
- 2. FilterChainProxy :
 - It is an interface that defines how to manage and execute chains of security filters (AuthorizationFilter, UsernamePasswordAuthenticationFilter, CsrfFilter, ExceptionTranslationFilter etc) in spring security
 - It provides methods for accessing filter chains, determining the appropriate chain for a request and integrating with the application context



```
<!-- spring-security-core dependency -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
    <version>5.5.2</version>
</dependency>
```

```
<!-- spring-security-config dependency -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
    <version>5.5.2</version>
</dependency>
```

```
<!-- spring-security-web dependency -->
<dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
    <version>5.5.2</version>
</dependency>
```

```
<servlet>
    <servlet-name>myds</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>myds</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

```
<filter>
    <filter-name>springSecurityFilter</filter-name>
    <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
    <filter-name>springSecurityFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
<authentication-manager>
    <authentication-provider>
        <password-encoder hash="bcrypt" />
        <user-service>
            <user name="admin" password="$2a$12$iePHJJ8K6jXFLKtOz/asiub0scSE5RDPm3ePMkPAI" />
        </user-service>
    </authentication-provider>
</authentication-manager>
```

eclipse-workspace-web-module - SpringSecurityXml3/src/main/webapp/WEB-INF/spring-security.xml - Eclipse IDE

File Edit Source Source Navigate Project Run Window Help

Project Explorer MyControllers.java contact-us.jsp user-panel.jsp admin-panel.jsp spring-security.xml

```
1<beans xmlns="http://www.springframework.org/schema/beans"
2  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  xsi:schemaLocation="http://www.springframework.org/schema/beans
4      http://www.springframework.org/schema/beans/spring-beans.xsd
5      http://www.springframework.org/schema/security
6      http://www.springframework.org/schema/security/spring-security.xsd">
7
8
9
10 <http auto-config="true">
11     <form-login/>
12     <intercept-url pattern="/adminPanel" access="hasRole('ROLE_ADMIN')"/>
13     <intercept-url pattern="/userPanel" access="hasRole('ROLE_USER')"/>
14 </http>
15
16
17 <authentication-manager>
18     <authentication-provider>
19         <password-encoder hash="bcrypt"/>
20         <user-service>
21             <user name="admin" password="$2a$12$iePHJJ8K6jXFLKtOz/asiub0scSE5RDPm3ePMkP4"/>
22             <user name="user" password="$2a$12$UzZzyAQKpr0CMIQTAOEV5uQXxc7ne4wBX1r52R4E1"/>
23         </user-service>
24     </authentication-provider>
25 </authentication-manager>
26
27 </beans>
```

Design Source

Writable Smart Insert 14:71:710

Windows Taskbar Icons

```
<input type="hidden" name="${tokenName}" value="${_
```

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3
4 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <meta charset="ISO-8859-1">
10 <title>Insert title here</title>
11 </head>
12 <body style="background-color: yellow;">
13     <h3> Login Page </h3>
14
15     <form action='<c:url value="/Login" />' method="post">
16         Username : <input type="text" name="username" /> <br/> <br/>
17         Password : <input type="password" name="password" /> <br/> <br/>
18         <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
19         <input type="submit" value="Login" />
20     </form>
21 </body>
22 </html>
```

```
        <rule>
            <!-- Admin Panel -->
            <url-pattern>/adminPanel</url-pattern>
            <login-page>/loginPage</login-page>
            <login-processing-url>/login</login-processing-url>
            <authentication-failure-url>/errorPage</authentication-failure-url>
            <access>hasRole('ROLE_ADMIN')</access>
        </rule>
```

```
<!-- jstl dependency -->  
<dependency>  
    <groupId>javax.servlet</groupId>  
    <artifactId>jstl</artifactId>  
    <version>1.2</version>  
</dependency>
```

eclipse-workspace-web-module - SpringSecurityJava3/src/main/webapp/WEB-INF/views/home.jsp - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer home.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10    <h2> Welcome to Smart Programming </h2>
11
12    <a href="home"> Open Home Page </a> <br/> <br/>
13    <a href="contactUs"> Open Contact Us Page </a> <br/> <br/>
14
15    <a href="adminPanel"> Open Admin Panel </a> <br/> <br/>
16    <a href="userPanel"> Open User Panel </a> <br/> <br/>
17 </body>
18 </html>
```

Java Resources

- src/main/java
- src/main/resources
- Libraries

Deployed Resources

src

- main
 - java
 - resources
- webapp
 - WEB-INF
 - views
 - admin-panel.jsp
 - contact-us.jsp
 - home.jsp
 - user-panel.jsp

index.jsp

target

pom.xml

SpringSecurityXml1

SpringSecurityXml2

SpringSecurityXml3

SpringSecurityXml4

SpringSecurityXml5

SpringWebAppJavaConfig1

SpringWebAppJavaConfig2

SpringWebAppJavaConfig3

SpringWebDownloadDemo1

src/main/resources - SpringSecurityJava3

Windows Taskbar

Eclipse IDE Bottom Bar

```
Project Explorer X home.jsp WebSecurityConfig.java X
> JspIncludeDemo
> Servers
> SpringMvcTiles1
> SpringMvcTiles2
> SpringMvcTiles3
> SpringSecurityJava1
> SpringSecurityJava2
> SpringSecurityJava3
> Deployment Descriptor: SpringSecurityJava3
> JAX-WS Web Services
> Java Resources
  > src/main/java
    > in.sp.configurations
    > in.sp.controllers
    > in.sp.security
      > SecurityWebAppInitializer.java
      > WebSecurityConfig.java
  > src/main/resources
  > Libraries
  > Deployed Resources
> src
  > main
    > java
    > resources
  > webapp
    > WEB-INF
      > views
        admin-panel.jsp
        contact-us.jsp
        home.jsp
        user-panel.jsp
      index.jsp
  > target
  pom.xml
> SpringSecurityXml1
> SpringSecurityXml2
> SpringSecurityXml3
> SpringSecurityXml4
20 {
21  @Override
22  protected void configure(HttpSecurity http) throws Exception
23  {
24      http
25          .authorizeRequests()
26              .antMatchers("/", "/home", "/contactUs").permitAll()
27              .antMatchers("/adminPanel").hasRole("ADMIN")
28              .antMatchers("/userPanel").hasRole("USER")
29              .and()
30          .formLogin();
31  }
32
33  @Bean
34  @Override
35  protected UserDetailsService userDetailsService()
36  {
37      UserDetails ud1 = User.withUsername("admin").password("$2a$12$XlY8ye1g8HpHUYl8CkCeS0").roles("ADMIN").build();
38      UserDetails ud2 = User.withUsername("user").password("$2a$12$u5LnyGriYCouGNKYnsJHFU").roles("USER").build();
39
40      InMemoryUserDetailsManager manager = new InMemoryUserDetailsManager();
41      manager.createUser(ud1);
42      manager.createUser(ud2);
43 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure. The `src/main/java` folder contains packages like `in.sp.configurations`, `in.sp.controllers`, and `in.sp.security`. The `WEB-INF/views` folder contains JSP files: `admin-panel.jsp`, `error-page.jsp`, `home.jsp`, and `login-page.jsp`.
- Editor View:** Displays the Java code for `WebSecurityConfig.java`. The code implements `WebSecurityConfigurerAdapter` and configures `HttpSecurity` to handle login requests for the `/adminPanel` endpoint.
- Bottom Status Bar:** Shows "Writable" and "Smart Insert" status, and a file count of "30 : 48 [9]".
- Bottom Taskbar:** Includes icons for Windows, search, file, Xd, Pr, and various browser tabs.

```
14
15 //in replacement of spring-security.xml file
16
17 @EnableWebSecurity
18 @ComponentScan("in.sp.controllers")
19 public class WebSecurityConfig extends WebSecurityConfigurerAdapter
20 {
21     @Override
22     protected void configure(HttpSecurity http) throws Exception
23     {
24         http
25             .authorizeRequests()
26                 .antMatchers("/adminPanel").hasRole("ADMIN")
27                 .and()
28             .formLogin()
29                 .loginPage("/login")
30                 .loginProcessingUrl("/loginForm")
31                 .failureUrl("/error");
32     }
33
34     @Bean
35     @Override
36     protected UserDetailsService userDetailsService()
37     {
```

eclipse-workspace-web-module - SpringSecurityJava4/src/main/webapp/WEB-INF/views/login-page.jsp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer WebInitializer.java WebConfig.java index.jsp MyController.java home.jsp WebSecurityConfig.java admin-panel.jsp login-page.jsp error-page.jsp

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body style="background-color: aqua;">
10    <h3> Secured Login Form </h3>
11
12    <form action="LoginForm" method="post">
13        Username <input type="text" name="username" /> <br/> <br/>
14        Password : <input type="password" name="password" /> <br/> <br/>
15        <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
16        <input type="submit" value="Login" />
17    </form>
18 </body>
19 </html>
```

Writable Smart Insert 12:28:307

File Explorer Task View Properties Run Debug Help