Some Important Terms in Spring Security

Last Updated : 23 Jul, 2025

Spring Security is a powerful **authentication and authorization** framework used to **secure Java-based web applications.** It integrates easily with Spring Boot and provides advanced security mechanisms such as OAuth2, JWT authentication, role-based access control, and protection against threats like CSRF, session fixation, and brute-force attacks. With the latest updates in **Spring Security 6.4**, developers can apply enhanced authentication strategies, improved method security, and modern security standards. In this article, we will learn the core security terminologies for effectively securing applications.

Important Terminologies in Spring Security

Some important terminologies in Spring Security are as follows:

1. Authentication
2. Authorization
3. Filter

Authentication

Authentication verifies the user's identity before granting access to the system. If authentication is successful, the request is processed, and a response is returned to the client.

Some authentication methods include:

- **Login Form Authentication**: It is a web page where users enter a username and password to gain access to secured resources.
- **HTTP Authentication:** In this, the server requests authentication credentials (username and password) from the client via HTTP headers.
- **Custom Authentication Method**: Using this method, developers can implement custom authentication logic using AuthenticationProvider and UserDetailsService.
- **Passkeys Support (New in Spring Security 6.4):** This is a passwordless authentication mechanism using cryptographic keys instead of traditional passwords.
- **One-Time Token Authentication (New in Spring Security 6.4)**: This is a temporary authentication mechanism providing enhanced security for sensitive operations.

Authorization

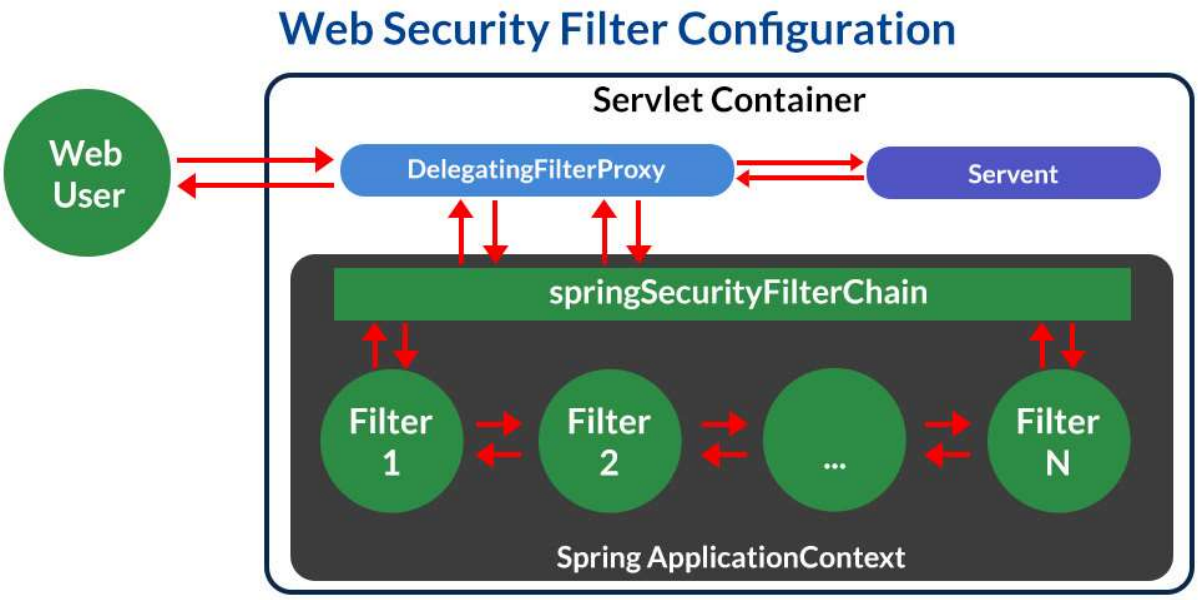
Authorization determines the level of access granted to authenticated users.

Some key authorization mechanisms include:

- **Access Control for URLs**: It restricts access to specific resources based on user roles using Spring Security's requestMatchers() API.
- **Secure Objects and Methods**: This uses annotations like @PreAuthorize and @PostAuthorize to enforce security at the method level.
- **Access Control Lists (ACLs)**: It defines permissions for specific users and roles, providing fine-grained authorization control.
- **Simplified OAuth 2.0 Configuration (Updated in Spring Security 6.4)**: It improves integration with third-party login providers like Google, GitHub, and Microsoft.
- **New Method Security Annotations (Updated in Spring Security 6.4)**: This introduces enhanced annotations to enforce role-based security policies at the method level.

Filter

Spring Security filters process security-related tasks during request handling. The filter chain executes different filters based on application needs.



- **Authentication Filter**: This handles user login and verifies credentials before granting access.
- **Authorization Filter**: This checks user permissions before allowing access to requested resources.
- **CSRF Protection Filter**: It prevents cross-site request forgery (CSRF) attacks by validating request tokens.
- **Session Management Filter:** It protects against session fixation attacks by enforcing secure session handling.
- **Refreshable SAML 2.0 Asserting Parties (New in Spring Security 6.4):** This enhances SAML 2.0 authentication by supporting dynamic metadata updates.
- **Security Observations for Filter Chain (New in Spring Security 6.4):** This introduces monitoring capabilities to track security events and filter execution.

Comment More info Advertise with us