SPRING SECURITY

Spring Security is a Java/Java EE framework that provides authentication, authorization and other security features

for enterprise applications

Spring Security is a powerful and highly customizable authentication and access-control framework

**5 SPRING SECURITY FEATURES**

* Authentication
* Authorization
* Principle
* Grant Authority
* Roles

CSRF= Cross-Site Request Forgery

CSRF (Cross-Site Request Forgery) is a security feature designed to protect users from unauthorized actions

on web applications where they are authenticated.

It primarily applies to stateful applications (like those using cookies for authentication).

**OAUTH2**

1. OAuth 2.0 is an authorization framework that allows third-party applications to obtain limited access to a user's

resources on a server, without exposing the user's credentials.

2. OAuth 2.0 allows users to grant applications access to their data without sharing their passwords.

The framework ensures security while enabling third-party applications to interact with user resources efficiently

**Basic Flow of OAuth 2.0**

* **User Requests Access:** The user tries to log into a client application (like a third-party app) using their credentials from another service (like Google or Facebook).
* **Redirect to Authorization Server**: The client redirects the user to the authorization server, asking for authorization. This typically involves a URL that includes parameters such as the client ID, redirect URI, and requested scopes.
* **User Grants Permission:** The user logs in to the authorization server and is prompted to grant permission to the client application.
* **Authorization Code**: If the user grants permission, the authorization server redirects the user back to the client application with an authorization code.
* **Access Token Request**: The client exchanges the authorization code for an access token by making a request to the authorization server.
* **Access Token Issued**: The authorization server validates the authorization code and returns an access token (and optionally a refresh token).
* **Access Resources:** The client can now use the access token to make requests to the resource server to access the user's resources.

**Spring IoC** (Inversion of Control) Container is the core of Spring Framework. It creates the objects, configures and assembles their dependencies, manages their entire life cycle. The Container uses Dependency Injection(DI) to manage the components that make up the application

| **Feature** | **Spring Framework** | **Spring Boot** |
| --- | --- | --- |
| Purpose | General-purpose framework | Simplified setup for Spring applications |
| Configuration | Extensive configuration required | Auto-configuration with sensible defaults |
| Dependency Management | Manual management of dependencies | Starter dependencies for easy management |
| Deployment | Typically WAR files for servlet containers | Standalone JAR with embedded server support |
| Development Speed | Slower due to configuration | Faster due to auto-configuration |
| Learning Curve | Steeper learning curve | Easier for beginners |