**SPRING SECURITY**

Spring Security is a powerful and customizable authentication and access control framework for securing Java applications.

It is widely used in Spring-based applications to handle tasks such as user authentication, authorization, role-based access control, and securing web applications or REST APIs.

Key Concepts of Spring Security:

***Authentication:***

* Verifying who the user is.
* Spring Security provides support for various authentication methods, including form-based login, basic authentication, OAuth, JWT, and custom authentication mechanisms.

***Authorization(Access Control):***

* Determining what actions an authenticated user is allowed to perform. This is typically based on roles or permissions.
* Spring Security helps in defining role-based or permission-based access control to methods, endpoints, or resources.

***Security Filters:***

* Spring Security operates using a filter chain where multiple filters like for login, logout, authentication intercept incoming requests and handle security concerns.
* These filters are customizable.

***CSRF Protection:***

* Cross-Site Request Forgery (CSRF) protection is enabled by default in Spring Security to prevent malicious actions being taken on a user's behalf without their consent.

***Password Encoding:***

* Spring Security provides support for password hashing and encoding mechanisms to securely store passwords.
* Passwords can be hashed using algorithms like BCrypt, Pbkdf2, etc.

**Dependency Required:**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

**SPRING SECURITY FOR USER AUTHENTICATION:**

[**LalithaRavuri/SpringSecurity**](https://github.com/LalithaRavuri/SpringSecurity)