



Spring Security

Core Spring Security Components

Core Spring Security Component

1. UserDetails

The **UserDetails** interface represents a user in the Spring Security framework. It provides methods to get user information such as username, password, and authorities.

Purpose: To encapsulate user information, including authentication and authorization details.

Implementation: You can use it to extend your **User Entity**.

Core Spring Security Component

2. UserDetailsService

The **UserDetailsService** interface is a core component in Spring Security that is used to retrieve user-related data. It has a single method: **loadUserByUsername**

Purpose: To fetch user details from a datasource (e.g., database) based on the username.

Implementation: You typically implement this interface to load user details, such as username, password, and roles, from your own user repository.

Core Spring Security Component

3. InMemoryUserDetailsManager

The **InMemoryUserDetailsManager** is a Spring Security provided implementation of **UserDetailsService** that stores user information in memory.

Purpose: To store user details in memory, typically for testing or small applications. You define users directly in the configuration.

Core Spring Security Component

3. PasswordEncoder

The **PasswordEncoder** interface is used for encoding and validating passwords. It has methods for encoding raw passwords and matching encoded passwords.

Purpose: To securely hash passwords before storing them and to verify hashed passwords during authentication.

Common Implementations:

- **BCryptPasswordEncoder**
- **Pbkdf2PasswordEncoder**
- **SCryptPasswordEncoder**

