**Spring Security**

**WHAT IS SPRING SECURTY**

* A framework that focuses on providing authentication and authorization mechanisms to spring application
* It is a library that can be used, extended to customize as per the programmer’s needs.
* We can define our own authentication process which can range from basic authentication using a username and a password to a complex one such as two-factor authentication using tokens and OTP’s. Also, we can use various databases – both relational and non-relational, use various password encoders, lock malicious users out of their accounts, and so on.
* Spring Security operates in two major areas: Authentication and Authorization.

Authentication

* Authentication means that, while accessing certain restricted resources, the user actually is the right person to do so. There are two processes to make sure that the user is authentic: identification and verification. For example, a user is authenticated through their username and password, which is typically stored in a database.

Authorization

* Authorization determines the extent of a user’s right to access restricted resources. It ensures that a user is allowed to access only those parts of the resource that one has been authorized to use. The user roles come as part of the authorization.

**BASIC COMPONENTS OF SPRING SECURITY**

* AuthenticationFilter
* AuthenticationManager
* AuthenticationProvider
* UserDetailsService
* PasswordEncoder
* Spring Security Context
* Form Login
* Login with a Database
* Login Attempts Limit

**How to enable spring security on the spring boot application?**

* Enabling spring security on a spring boot application is so simple. Just add the spring security dependency in the pom.xml

<dependency>

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

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

</dependency>

* After adding the security dependency on the spring boot application. Now the system authenticates all the request which are coming to the application.
* If you request a rest API once after you added the spring-boot-starter-security dependency, the system returns an unauthorized response.

**How to override the default configuration on spring security?**

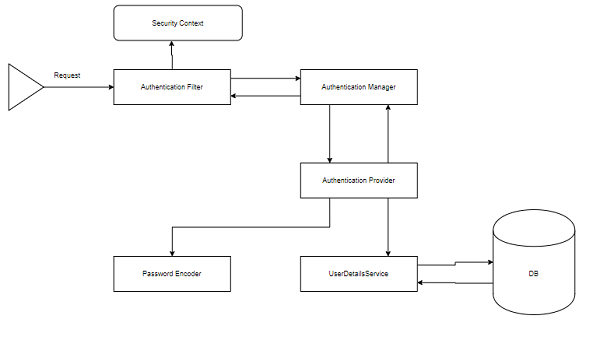
* It is able to override the default configuration of spring security by extending the class WebSecurityConfigurerAdapter.
* The WebSecurityConfigurerAdapter is an abstract class but it doesn't contain any abstract method but contains default configurations.
* The advantage of extending the adapter class is that we can configure Web security by overriding only those parts that we need. You want to override the configure method in the adapter class for your specific custom configuration. There are three variations of the configure method that we can override to configure and secure the application:
  + void configure( AuthenticationManagerBuilder auth): It is used to configure user details services. Overriding this method it is able to configure the application user source.
  + void configure( HttpSecurity http): It is used to configure how requests are secured by interceptors. Here you can configure the HTTP request authentication detail.
  + void configure( WebSecurity web): It is used to configure Spring Security’s filter chain. Here it is able to configure the web page configuration.

**PURPOSE OF SPRING SECURITY**

* Spring Security is the primary choice for implementing application-level security in spring applications.
* Generally, its purpose is to offer you a highly customizable way of implementing authentication, authorization, and protection against common attacks.

**THE ARCHITECTURE OF SPRING SECURITY**

* It starts with servlet filters.
* These filters intercept requests, perform operations on them, and then pass the requests on to next filters in the filter chain or request handlers or block them if they do not meet certain conditions.
* It is during this process that Spring Security can authenticate requests and perform various authentication checks on the requests.
* It can also prevent unauthenticated or malicious requests from accessing our protected resources by not allowing them to pass through.
* Thus our application and resources stay protected.

****

**SPRING SECURITY FEATURES**

* LDAP (Lightweight Directory Access Protocol)
* Single sign-on
* JAAS (Java Authentication and Authorization Service) LoginModule
* Basic Access Authentication
* Digest Access Authentication
* Remember-me
* Web Form Authentication
* Authorization
* Software Localization
* HTTP Authorization