Spring Security

Handles common vulnerabilities
- session fixation
- clickjacking
- click site request forgery

what spring security can do

- username / password
- SSO, Okta, LDAP
- App level authorisation OAuth
- microservice security (tokens JWT)
- methode level security

3 core concepts in spring security
- Authentication
- principal
- Roles
- Authorisation
- Granted authority

Principal - currently logged in user.

Granted authority - of the user is authorised then
spring grants authority (permission)

Roles - a group of authority - corse grained
permissions

Spring-boot-starter security

The maven dependancy that allows to add
security to spring boot application.

when the spring-boot-starter-security is added it will assign a login setup for the application

~~Fitters~~
Fulters-
Spring Security maintains a filter chain internally where each one of the filter has ~~th~~ a particular responsibility and filters are added or removed from the configuration

By default Spring Security
- adds mandatory authentication for URLs
- Adds login form
- Handles login error
- Creates a user & sets a default password.

How to ~~add a security~~ configure
Authentication manager builder.

- create a class that extends WebSecurityConfigure.
- add an override methode (configure(auth manager Builder

Then we have to specify what type of authentication we want
eg: InMemoryAuthentication
Here we have to pass user credential & role

```
auth.InmemoryAuthentication()
        .withUser("Neeraj")
        .password("12345")
        .Roles("USER");
```

add this above class
@EnableWebSecurity

we dont want to store the password as a string but an encoded text (hashed password)

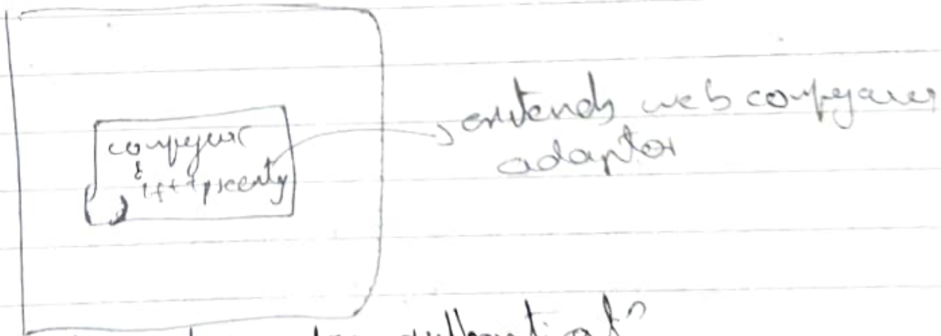
How to set password encoder
create a bean of type password encoder

@Bean
public PasswordEncoder getPasswordEncoder() {
    return NoOpPasswordEncoder.getInstance();

Authorisation in Spring

The authorisation allows to serve different API with different permissions. user api has certain page acces Admin has some other etc.
for that we create 3 APIs

Using an object of type HttpSecurity
configure paths & access restriction.



or use the same class for authentication
we create a class that extends websecurity configurer adapter & ~~create a~~ methode override a methode configure to get Http security object

there is http.authorizeRequest()
specify the mapping for path to permission

.antMatchers("/**").hasRole("USER")
all url to be accessible by user role

.hasAnyRole("user", "admin")

.antMatchers("/", "~~static~~/css", "~~static~~/js").permitAll()

when we give permission to the apis

go from most restricted url to least restricted

JWT
Jason Web Tokens. (JWT)

create a standard way for 2 parties to communicate
securely RFC7519.

authorisation techniques. using tokens
• session tokens - Reference token
• JSON web tokens (JWT) - value token

3 parts to JWT
header
payload
signature

The header & payload are visible and the signature is a string that can only be created by the server

The signature is used to validate the token.

JWT is not authentication methode it is used to remember the user.