**Spring Security:**

When we add spring security dependency, then automatically we get a login page and authentication is added for every end point in the application.

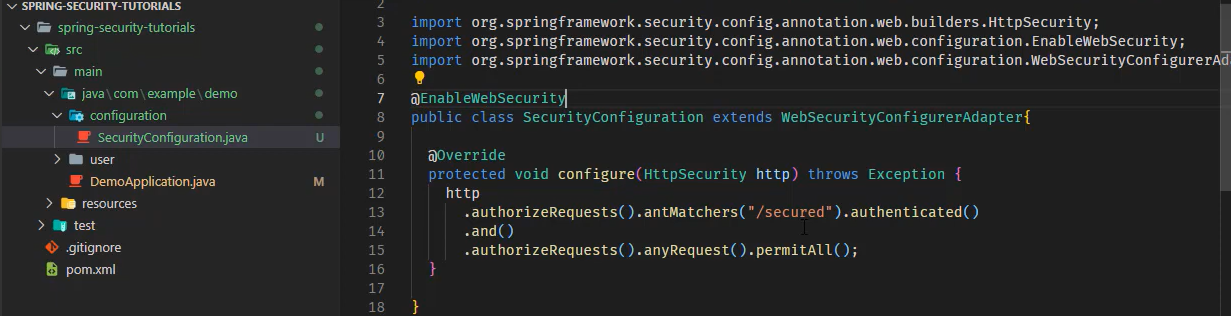
User name is user and password we can find in the Console.

Configuring the default spring security configuration:

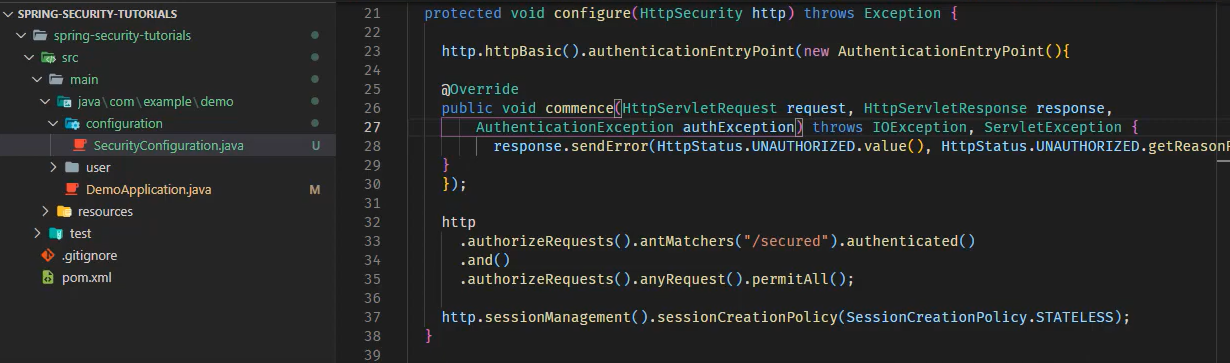
* We need to extend Web security Configure Adapter.
* Override configure with Http Security – to customize the authorizations of the application.

/secured - authenticated

/public - need not require login



* If we don’t want to get a form we need to use http.basic() line inside the configure method.
* Bu default we will have sessions stored in cookies, once we login (using basic auth) we can access anything directly when we use No auth in postman. It wont verify all the end points so we need to disable the session so that when we keep no auth in postman after keeping the basic auth it wont work.
* When we write http basic also, when we run in the browser it will ask login details in the alert box. This is because it adds header some headers. So we need to override the method which is adding headers. We remove the header line. So that we get only unauthorized message. It wont ask for login.
* By overriding commence method like this, we don’t get any login page. Just we get error message as unauthorized user.



* We can also add line csrf disable in order to overcome problem of forbidden error while using postman.

**User details Service**

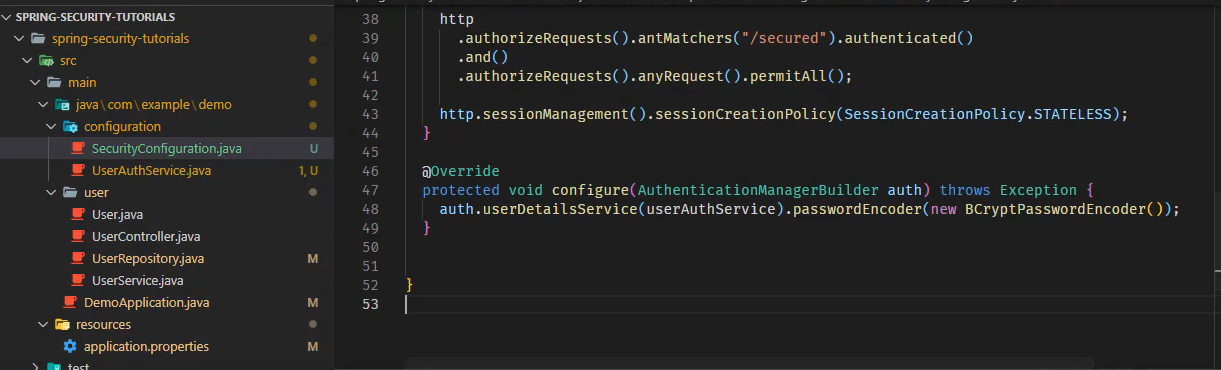
* *For user authentication, we need to implement the user details service. In order to override the current login form.*

Steps:

1. Create a class that implements the user details service and load by username method. Annotate this new class as @Service
2. Now, we need write the load by username method in the JPA class.
3. Use this JPA Class in the current class
4. Create a password encoder.
5. Create User Details object and return it.

Link: <https://www.youtube.com/watch?v=Jhtl2hxLEQU&list=PLv9DX4Sm66LCT-R3bOKxQ-k0c4uauTzl2&index=9>

* We have created new User Details class so we need to override the configure Authentication manager builder in order to override the security.
* We can use any encoder.



Other links:

JWT Implementation:

<https://www.kindsonthegenius.com/spring-boot/jwtjson-web-token-with-springboot-step-by-step-tutorial-part-1/>

<https://www.kindsonthegenius.com/spring-boot/jwtjson-web-token-with-springboot-step-by-step-tutorial-part-2/>

Medium Links:

<https://medium.com/geekculture/spring-security-authentication-process-authentication-flow-behind-the-scenes-d56da63f04fa>

<https://medium.com/codex/how-to-set-basic-authentication-for-api-endpoints-in-spring-boot-9cd8c79d26a2>