**How to enable spring security in our application**

**Fi**rst of all we will add the spring starter dependency to our project

The starter dependency will add a level of security to our application and every url will be intercepted by a filter and we can enter in only by entering the correct username and password.

|  |
| --- |
| <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-security</artifactId>  </dependency> |

|  |
| --- |
| server.port=2020  spring.security.user.name=foo  spring.security.user.password=foo |

**AUTHENTICATION**

**Authentication manager and authentication managerbuilder**

Authentication manager is the one that manages authentication for us.

It has an authenticate () method which return s the desired result of if a person is authenticated or not.

We do not use authentication manager directly rather we use a uthenticationmanager builder which buils the authentication object and holds it to authentication manager.

When we write a security configuration class then spring security thinks that I need not to work on my security system we can use the security one provided by the developer

|  |
| --- |
| @EnableWebSecurity  **public** **class** SecurityConfiguration **extends** WebSecurityConfigurerAdapter {    @Override  **protected** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception  {  auth.inMemoryAuthentication()  .withUser("blah")  .password("blah")  .roles("User");  }    @Bean  **public** PasswordEncoder getPasswordEncoder() {  **return** ~~NoOpPasswordEncoder~~.~~getInstance~~();  }  } |

We also will use a password encoder which will help us to encode the password and then save it inside the database.

**AUTHORIZATION**

A screenshot of a computer screen

Description automatically generated

We setup a controller for the same

|  |
| --- |
| @RestController  **public** **class** HomeResource {    @GetMapping("/")  **public** String home() {  **return**("<h1>Welcome</h1>");  }    @GetMapping("/user")  **public** String user() {  **return**("<h1>Welcome USer.both admin and user allowed</h1>");      }  @GetMapping("/admin")  **public** String admin() {  **return**("<h1>Welcome Admin,only admin allowed</h1>");  }  } |
| @Overrride  **protected** **void** configure(HttpSecurity http)**throws** Exception{  http.~~authorizeHttpRequests~~()  .antMatcher("/user").hasRole("/user")  .antMatcher("/admin").hasRole("/admin")  .antMatcher("/").permitAll()  .and().formLogin();  } |