**Spring Boot security with Angular front End**

How to enable spring boot security:

<dependency>

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

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

<scope>test</scope>

</dependency>

**Creating inMemoryUser**

****

In this we not using any encoding password is going to save as Plan text 

**Calling rest API from angular with authoniation in header**

let name="test";

    let password="test";

    let auth="Basic "+window.btoa(name+":"+password);

    let header=new HttpHeaders({

      Authorization:auth

    })

 this.http.get("http://localhost:8080/hello/Deepak",{headers:header}).subscribe(res=>{

      console.log(res)

    },erro=>{

      console.log(erro)

    });

****

**HttpIntercept and adding header authonication**

**service --------------🡪configuration in **

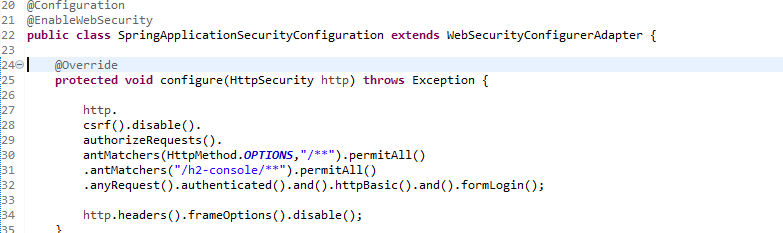
Config

 providers: [

      {provide:HTTP\_INTERCEPTORS,useClass:httpBaseInter,multi:true}

  ],

**How to whitelist url and enable h2-console(headers.frameOptions)**

****

****

**New user and DB get user from DB**

**Step 1**

@Autowired

**private** UserDetailsService userDetailsService;

@Bean

**public** AuthenticationProvider authProvider() {

DaoAuthenticationProvider provider=**new** DaoAuthenticationProvider();

provider.setUserDetailsService(userDetailsService);

provider.setPasswordEncoder(**new** BCryptPasswordEncoder());

**return** provider;

}



**Step2**

**Implement UserDetailsSerice**

****

**Step 3**

**Implementing userdetails**

****

**Step 4:**

**User role:**

**public** **class** UserRoleDTO **extends** BaseMapperDTO **implements** GrantedAuthority{

@Override

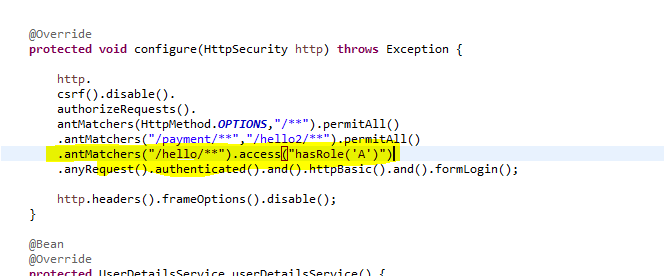
**public** String getAuthority() {

**return** **this**.getRole().name();

}

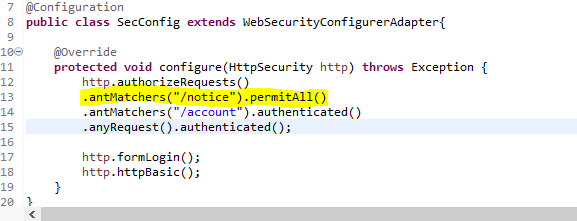
**Other class also like db entity , dbrepo etc**

**How to protect URL on based Role :**

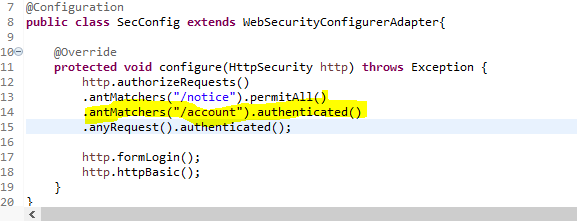
****

**Spring Boot Sercurity New Course**

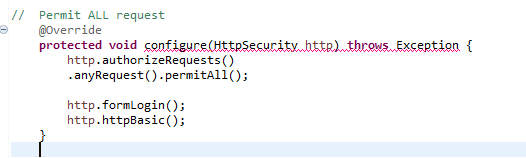
**How to permit specific request without authentication**

****

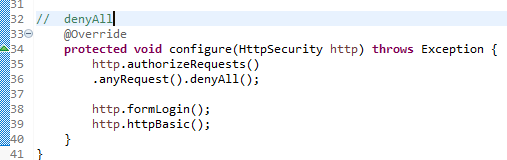
**How to authenticate specific request**

****

**How to permit all request.**

****

**Denay ALL request.**

****

**Custom success and failer login handler**

1. **Create a link which redirect to server login form.**
2. **Form Login**
3. **On success redirect back to home page of angular app.**

****

****

**JWT (JSON Web token)**

**Step 1** . Add dependence.

|  |
| --- |
| <dependency> |
|  | <groupId>javax.xml.bind</groupId> |
|  | <artifactId>jaxb-api</artifactId> |
|  | </dependency> |
|  | <dependency> |
|  | <groupId>io.jsonwebtoken</groupId> |
|  | <artifactId>jjwt</artifactId> |
|  | <version>0.9.1</version> |
|  | </dependency> |

**Step 2.**

**Create JWT utilie class**

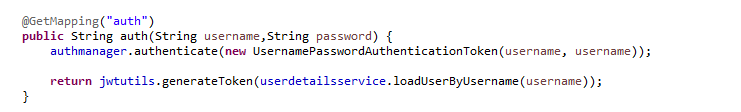
****

**Step 3. Create authentication config**

****

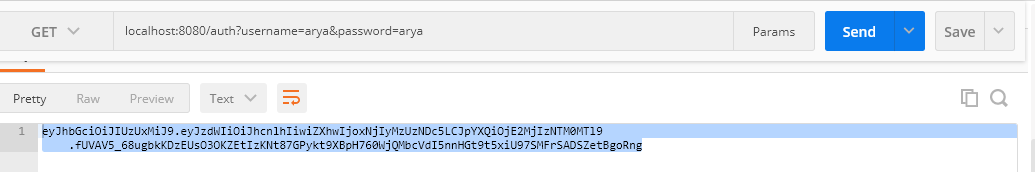
**Step 4. Generate token using authentication manager**

****

****

**Step 5. Create Filter.**

****

****

