**Spring Security + Spring Boot + Spring MVC**

AES user authentication and security is configured on every Spring Boot run from beans annotated with @Configuration. All users regardless of their credentials are allowed to access the login page **templates/login/index.html** and any unauthorized access to other views will redirect to this page. When a user submits a login form from index.html, the userApp Angular module handles the post request and begins the authentication process. A valid user login will generate a Principal instance and a set of tokens for the session which will be checked and validated at every page. Below are the relevant files for security implementation in the order they are called.

**templates/login/index.html**

**static/js/controllers/user/LoginCtrl.js**

When login post request is made, the login() function of Angular module userApp calls the Rest Controller endpoint /aes/login which will return a valid Principal user instance based on WebSecurityConfig configuration and CustomUserDetailService validation methods. When Principal user is returned to the Login controller of userApp, its authority is used in a switch case to forward the validated user to the correct landing page according to their role. If no valid Principal is returned, Angular will set bunkCreds = true, which will show an error message on the log-in page.

**com.revature.aes.config.WebSecurityConfig.java**

Spring Boot configuration bean where HttpSecurity, AuthenticationManagerBuilder, and PasswordEncoder are declared and defined. HttpSecurity configuration defines access rights to views for all user roles. Any invalid access of a view will redirect to the log-in page. AuthenticationManagerBuilder defines rules for user authentication using UserDetailService and encodes the password using PasswordEncoder().

**com.revature.aes.service.CustomUserDetailService.java**

Principal user is validated by CustomUserDetailService which finds the username in the database and returns an instance of User bean. A valid password is validated using the Security bean, and the User instance role is used to determine its authority. If all three are valid, a Principal User instance is returned.