Step 1: EnableWebSecurity Annotation

This annotation is used to enable Spring Security in the application. It allows the configuration class to be picked up during the application startup.

Step 2: SecurityConfig Class

This class extends the WebSecurityConfigurerAdapter class, which provides default security configurations and allows customization.

Step 3: getpassword() Method

This method is annotated with @Bean and creates a bean of type PasswordEncoder. In this case, NoOpPasswordEncoder.getInstance() is used, which means the passwords are stored in plain text without any encoding. Note that this approach is not secure and should not be used in production. It is used here for simplicity.

Step 4: configure() Method for Authorization

This method configures the authorization rules and security settings for different URL paths. It is also an overridden method from WebSecurityConfigurerAdapter. The following rules are defined:

Requests to "/access1" should have the "ADMIN" role.

Requests to "/access2" should have the "USER" role.

Requests to "/access3" should have either "ADMIN" or "USER" role.

The root URL ("/") is permitted to all users.

The formLogin() method is used to enable form-based authentication.