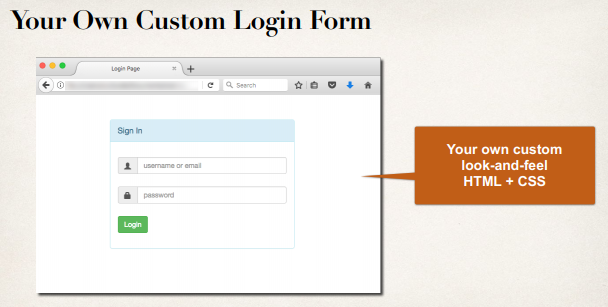
**48.3. Spring Security - Custom Login Form Overview**

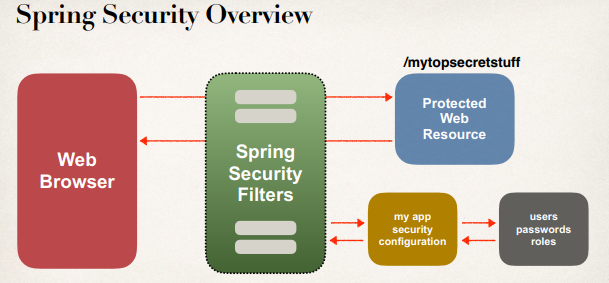
**Your Own Custom Login Form**:

Spring provide us default logging form but this is not ideal. Here we will create our own custom logging form. We can control the look and feel, use our own HTML, CSS, company logo and many more that we want.

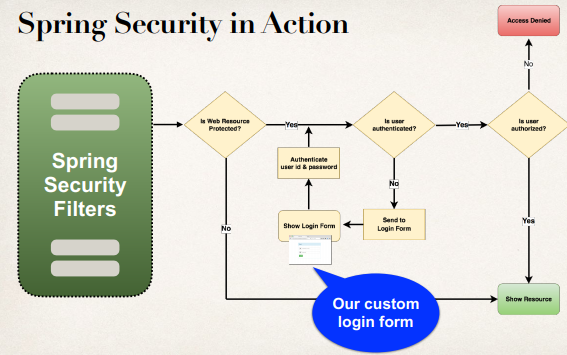


**Spring Security Overview**:

Here we have our web browser and we are trying to access some protected web resources. Here we have Spring security filters in place that will be actually kind of pre-process and post-process. They basically make use of their app configuration and read userid, password, and roles.



**Spring Security in Action**:



**Development Process (Step-by-Step)**:

1. Modify Spring Security Configuration to reference custom login form
2. Develop a Controller to show the custom login form
3. Create custom login form
   1. HTML (CSS optional)
   2. Spring MVC form tag **<form:form>**

**Step 1: Modify Spring Security Configuration**:

|  |  |
| --- | --- |
| Method | Description |
| configure(AuthenticationManagerBuilder | Configure users (in memory, database, ldap, etc) |
| configure(HttpSecurity) | Configure security of web paths in application, login, logout etc |

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http.authorizeRequests()

.anyRequest().authenticated()

.and()

.formLogin()

.loginPage("/showMyLoginPage")

.loginProcessingUrl("/authenticateTheUser")

.permitAll();

}

48.3. Spring Security - Custom Login Form Overview