

Lab 15

Introduction to Securing Web Application

Mục tiêu

- Implement cơ chế security cho web application phần
- Xác nhận authenticated user trước khi truy cập đến các trang khác của ứng dụng web

Phần I Bài tập step by step

Bài 1.1

Tạo Project Web và cấu hình để người dùng phải đăng nhập rồi mới được vào các trang khác.

STEP 1: Tạo project

- New Project -> Java Web -> Web Application -> Đặt tên "JspServlet_Lab015" -> Next -> Next -> Finish.

STEP 2: Tạo các trang .jsp

- Xóa file "index.xhtml" được tạo sẵn.
- Tạo file "index.jsp" có nội dung:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <center>
      <h1>WELCOME TO MY WEBSITE</h1>
    </center>
  </body>
</html>
```

- Tạo file "login.jsp" có form để login:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <center>
      <h1>LOGIN FORM</h1>
      <form action="j_security_check">
        <table>
          <tr>
            <td>User name:</td>
            <td><input type="text" name="j_username"/></td>
          </tr>
          <tr>
            <td>Password:</td>
            <td><input type="password" name="j_password"/></td>
          </tr>
          <tr>
            <td></td>
            <td>
              <input type="submit" value="Login"/>
              <input type="reset" value="Clear"/>
            </td>
          </tr>
        </table>
      </form>
    </center>
  </body>
</html>
```

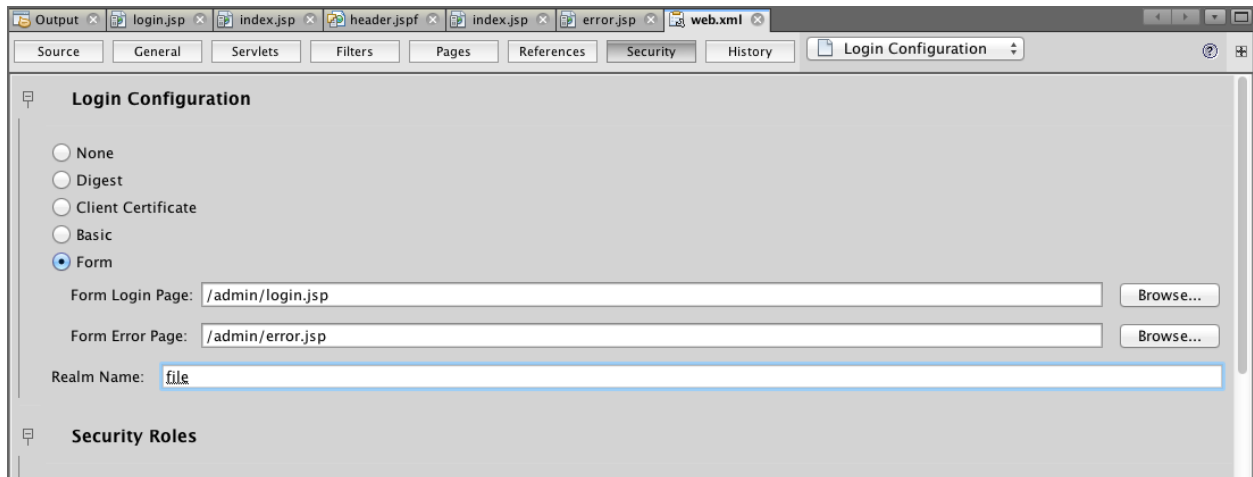
Các tên "j_security_check", "j_username", "j_password" là bắt buộc phải chính xác.

- Tạo file "loginError.jsp":

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <center>
      <h3 style="color: red">WRONG USER OR PASSWORD!</h3>
      <a href="login.jsp">Back to Login page</a>
    </center>
  </body>
</html>
```

STEP 3: Thêm Security Entries vào file web.xml

Mở file web.xml, dưới dạng interface -> Chọn tab Security -> Mở rộng Login Configuration -> Chọn Form -> Thêm các trang login.jsp và error.jsp đã tạo ở trên như sau:

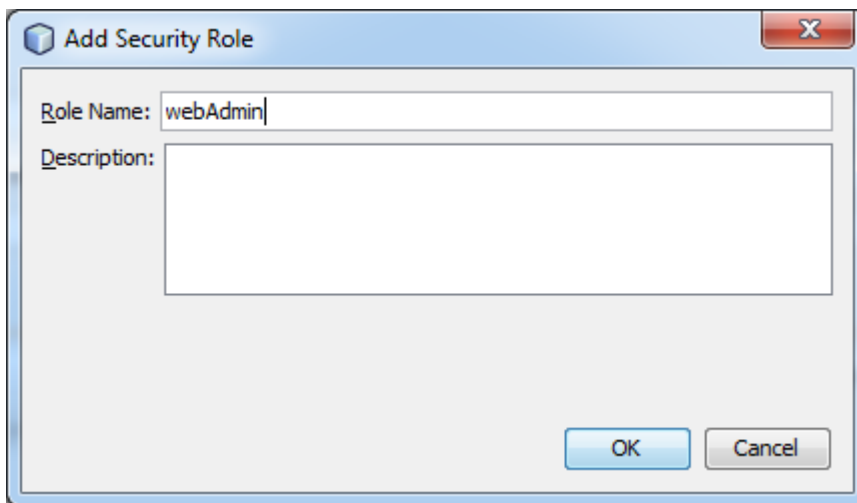
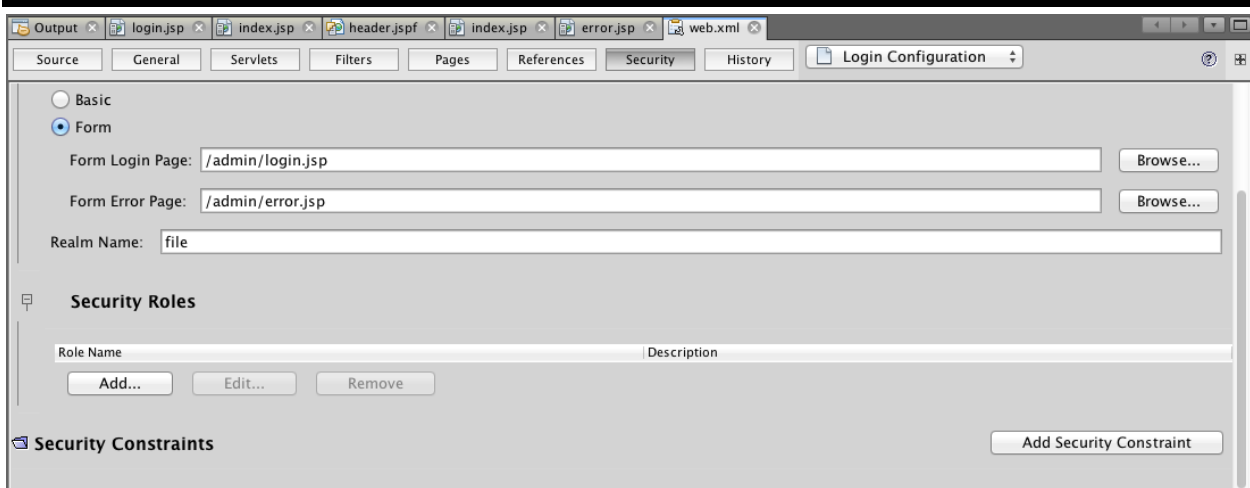


Trong source code của file web.xml sẽ thêm đoạn config sau:

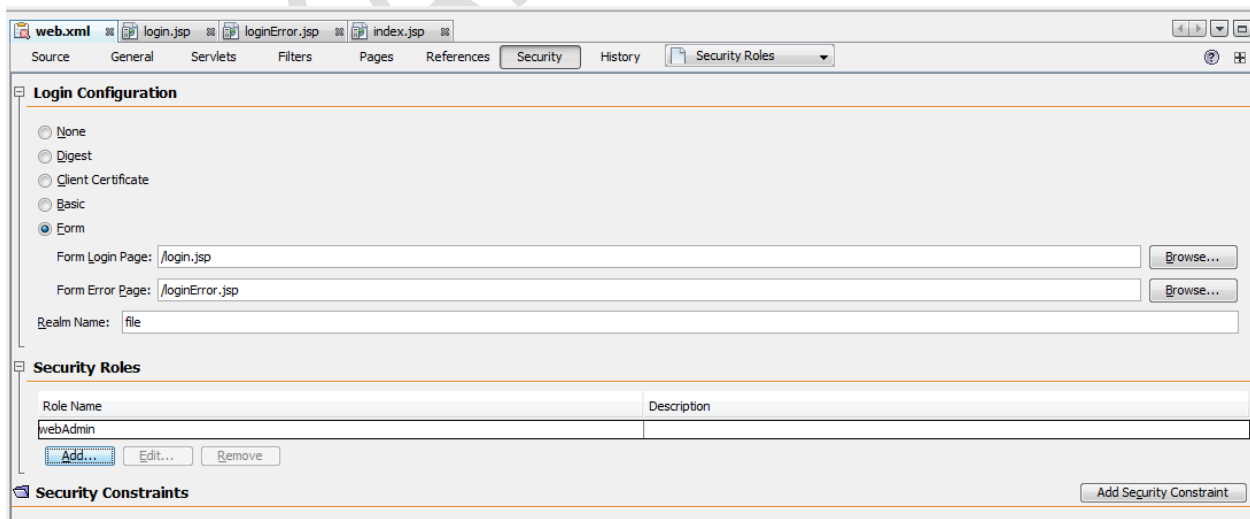
```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>file</realm-name>
    <form-login-config>
        <form-login-page>/admin/login.jsp</form-login-page>
        <form-error-page>/admin/error.jsp</form-error-page>
    </form-login-config>
</login-config>
```

Đoạn config này sẽ thông báo cho servlet container form-based authentication được sử dụng, các file login và error sẽ được gọi và check định danh người dùng.

Mở Security tab -> Mở rộng Security Roles -> Thêm role bằng click Add



Role Name: gõ webAdmin -> OK



Trong file web.xml, đoạn config tương ứng được sinh ra:

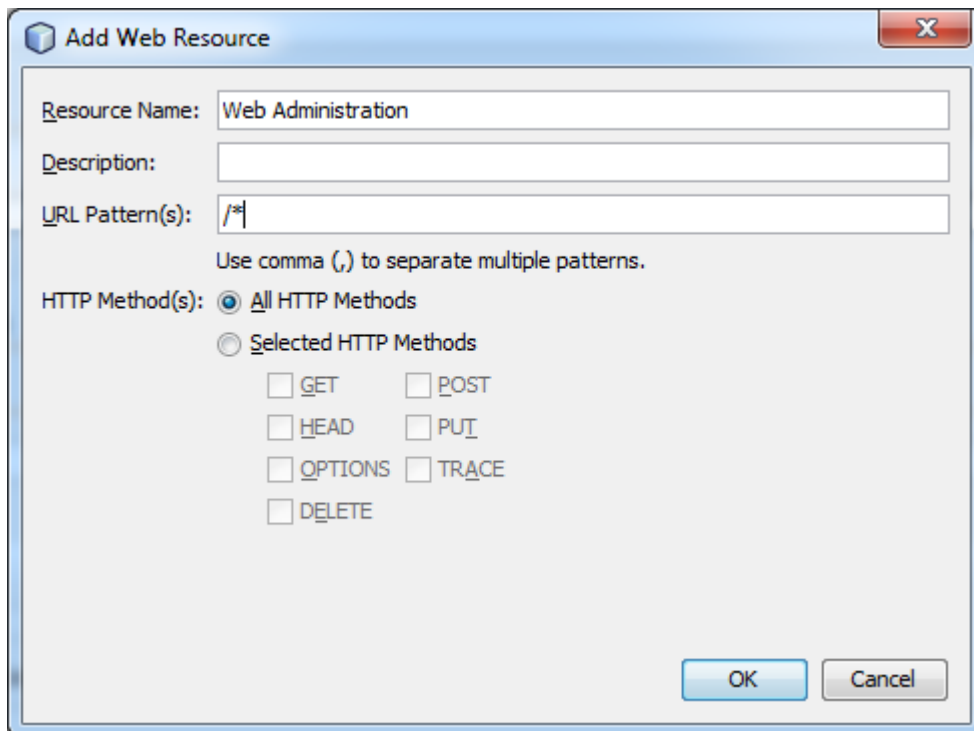
```
<security-role>
  <description/>
```

```
<role-name>webAdmin</role-name>  
</security-role>
```

Quay lại Security Tab -> Add Security Constraint

Display Name: gõ Admin

Web Resource Collection -> Add



Resource Name: gõ Web Administration

URL Patterns: /*

Chọn All HTTP Methods -> OK

Chọn Enable Authentication Constraint -> Edit

Admin [Remove]

Display Name:

Web Resource Collection:

Name	URL Pattern	HTTP Method	Description
eMarket Administration	/admin/*		

[Add...] [Edit...] [Remove]

☒ Enable Authentication Constraint

Description:

Role Name(s): [Edit]

☐ Enable User Data Constraint

Description:

Chọn Edit Role name của Authentication Constraint:

Edit Role Names [X]

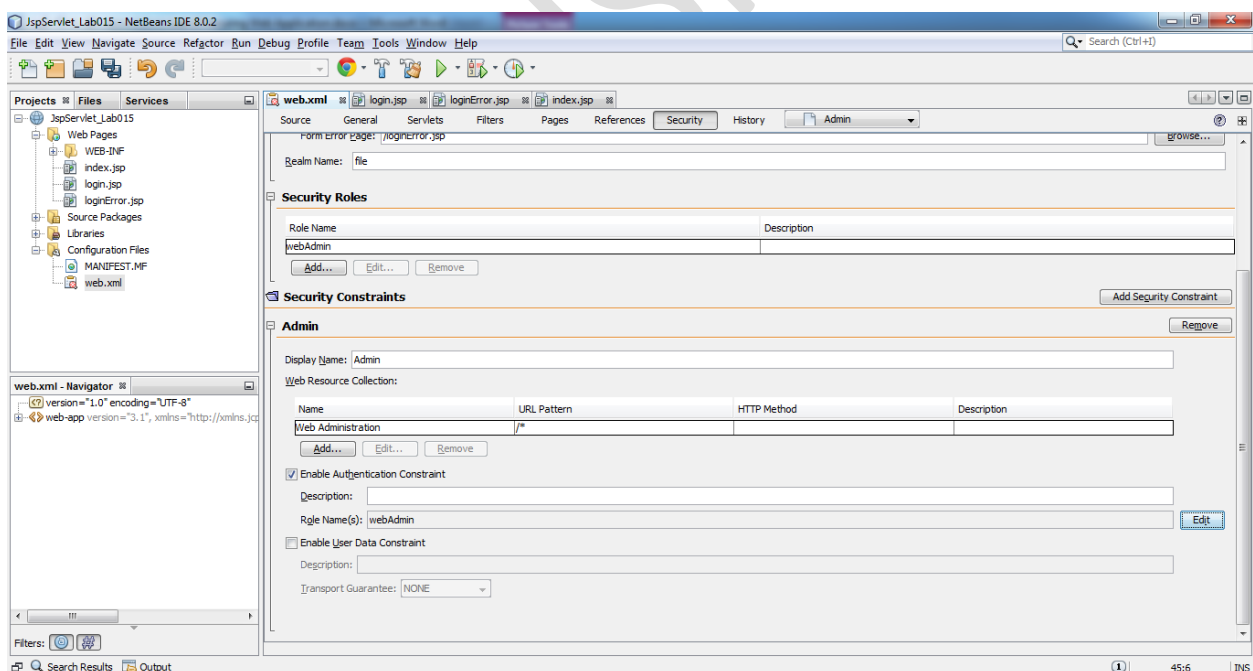
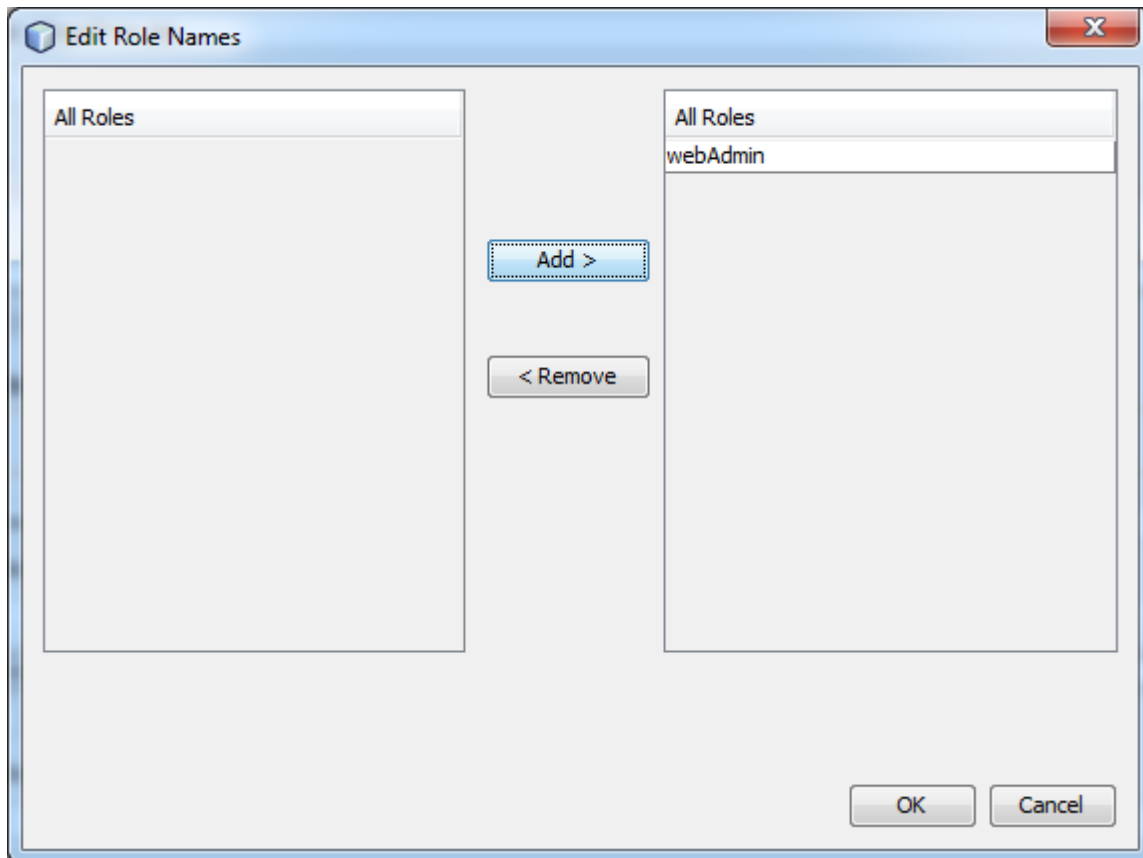
All Roles

webAdmin

All Roles

[Add >] [< Remove]

[OK] [Cancel]



Config tương ứng cho toàn bộ các bước đã làm trong file web.xml như sau:

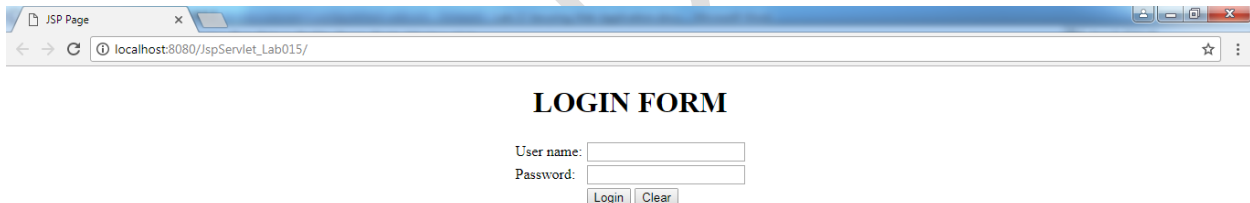
```
<security-constraint>
  <display-name>Admin</display-name>
  <web-resource-collection>
    <web-resource-name>Web Administration</web-resource-name>
```

```
<description/>
<url-pattern>/*</url-pattern>
</web-resource-collection>
<auth-constraint>
  <description/>
  <role-name>webAdmin</role-name>
</auth-constraint>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>file</realm-name>
  <form-login-config>
    <form-login-page>/login.jsp</form-login-page>
    <form-error-page>/loginError.jsp</form-error-page>
  </form-login-config>
</login-config>

<security-role>
  <description/>
  <role-name>webAdmin</role-name>
</security-role>
```

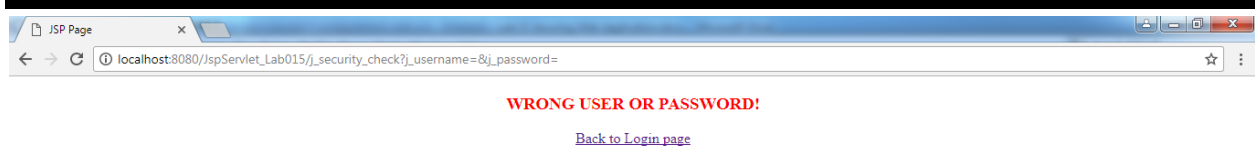
Với tất cả các truy cập đến các trang của ứng dụng web, chương trình sẽ tự động chuyển hướng đến trang /login.jsp.

Deploy và chạy project, ngay từ đầu người dùng đã phải nhập vào user và password:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/JspServlet_Lab015/'. The page content is a simple login form titled 'LOGIN FORM'. It contains two input fields: 'User name:' and 'Password:'. Below the password field are two buttons: 'Login' and 'Clear'.

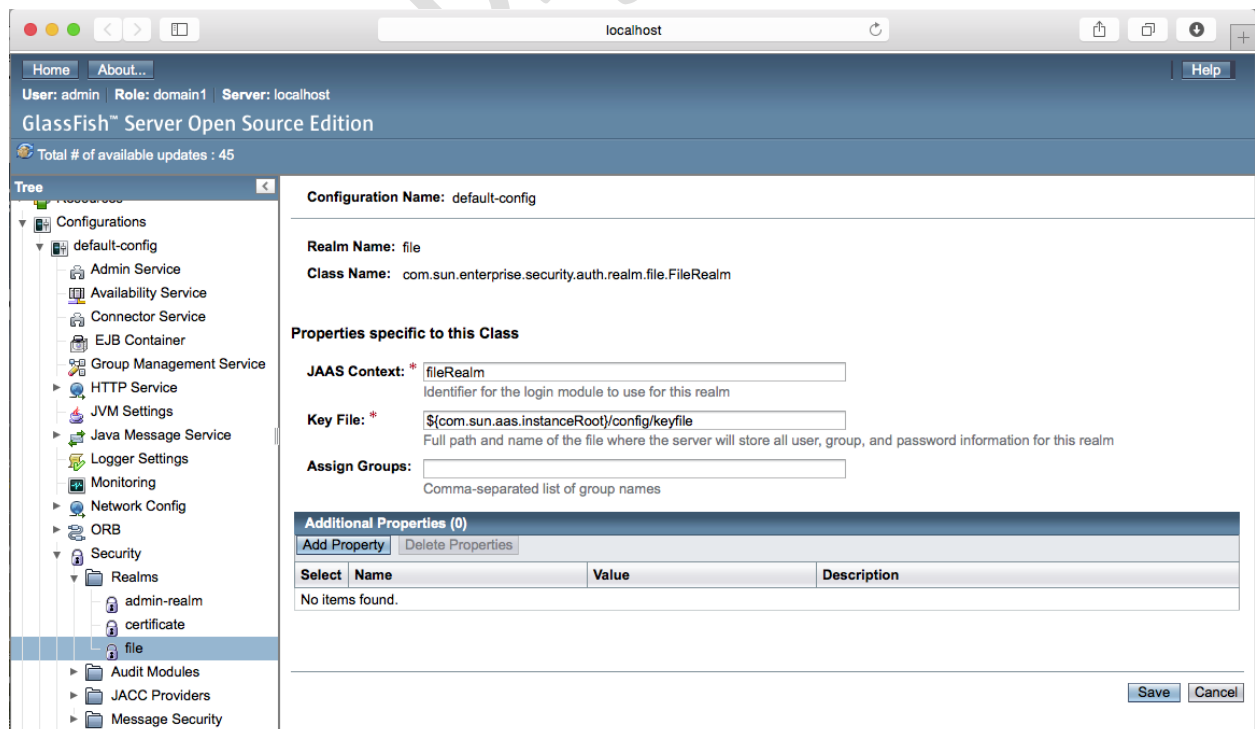
Submit form (click Login button) -> trang loginError.jsp sẽ được gọi.



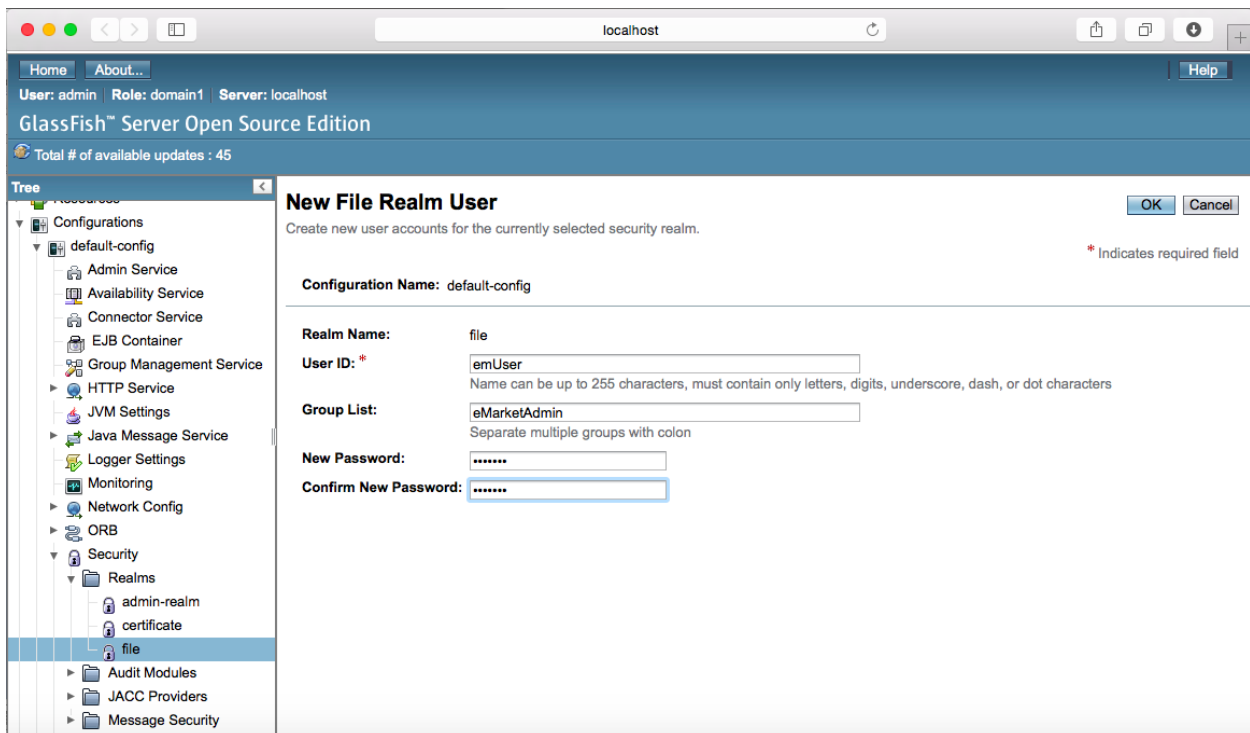
STEP 4: Thiết lập Users, Groups và Roles

(4.1) Tạo Users/Groups trên Server

Chọn Services Tab -> Servers -> GlassFish -> View Domain Admin Console -> Configuration -> default config -> Security -> Realms -> file



Ở dưới Edit Realm -> Chọn Manage Users -> Ở File Users click New



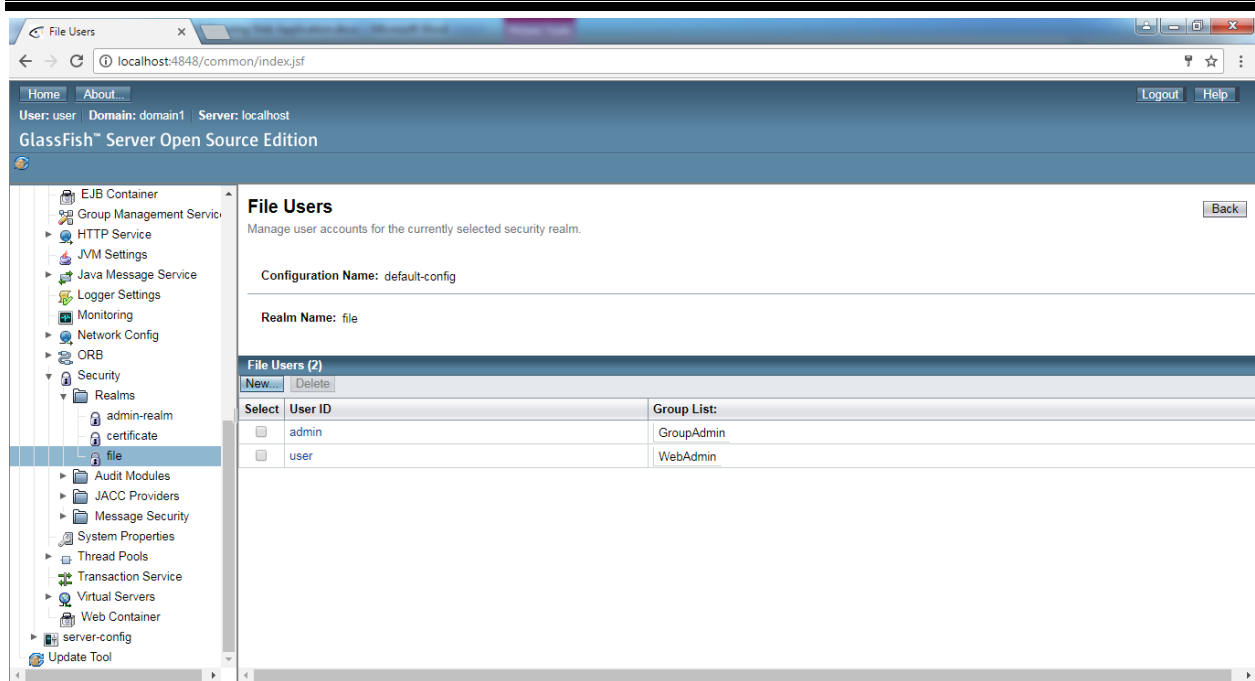
User ID: user

Group List: WebAdmin

Password: 1234\$

Confirm Password: 1234\$

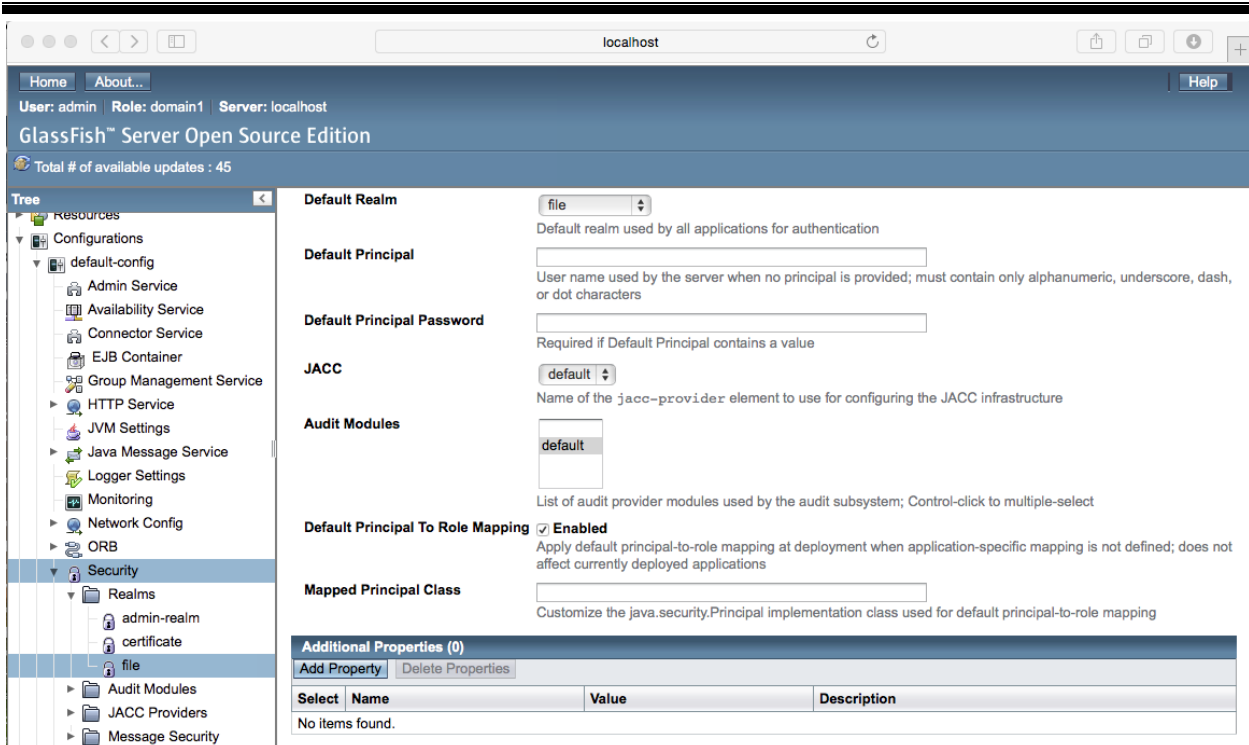
Chọn OK



(4.2) Map Roles cho User và Group

Ở step 2, chúng ta đã tạo form-based authentication bằng cách config file web.xml, tên role là WebAdmin. Chúng ta cần map group WebAdmin tạo ra ở step 4.1 với role tên WebAdmin ở trên.

Quay trở lại tab Security trên Admin Console của GlassFish -> Check Enabled cho Default Principal To Role Mapping -> Save.



Restart GlashFish để nó nhận các cấu hình chúng ta vừa thiết lập.

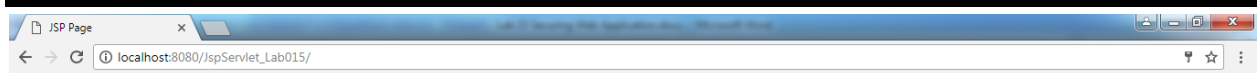
Deploy và chạy ứng dụng người dùng sẽ phải đăng nhập vào trước:



LOGIN FORM

User name:

Password:



WELCOME TO MY WEBSITE

Sau khi đăng nhập, trang /index.jsp sẽ hiện ra.

PHẦN 2: BÀI TẬP LÀM THÊM

Cùng thực hiện form-based authentication, thay vì thực hiện config trong web.xml, hãy sử dụng annotation `@ServletSecurity` và `@HttpConstraint`. Gợi ý, trong servlet xử lý của admin, thêm:

```
@ServletSecurity( @HttpConstraint(rolesAllowed =  
{"eMarketAdmin"}) )
```

HẾT