RESULTS WITH STRUTS 2 FRAMEWORK

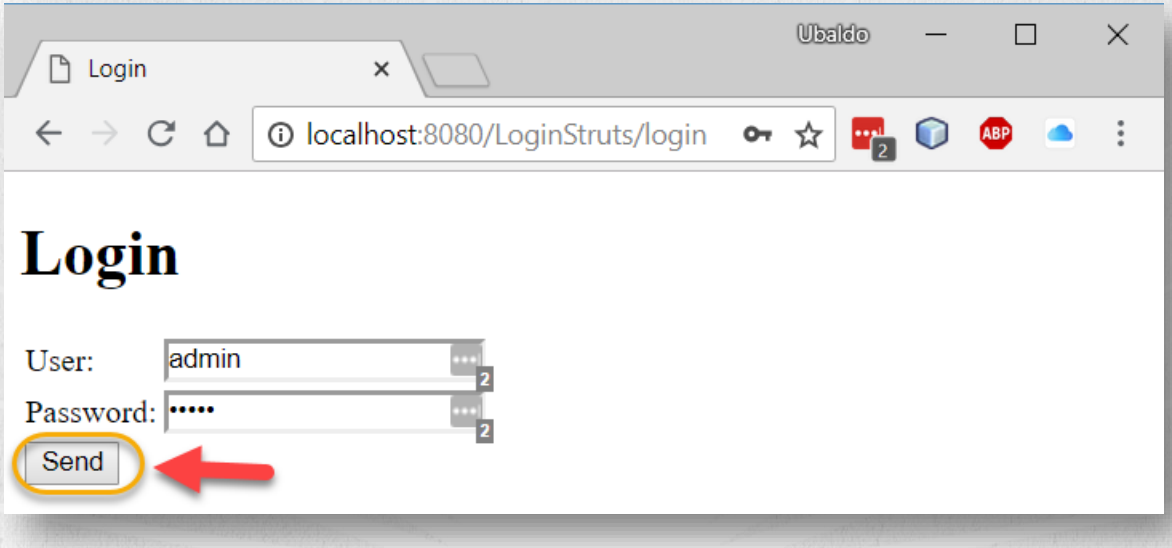
By the expert: Ubaldo Acosta

Global Mentoring

JAVA ONLINE COURSES UNIVERSITY

EXERCISE OBJECTIVE

Create an application to implement the use of Results with Struts2. At the end we should observe the following:



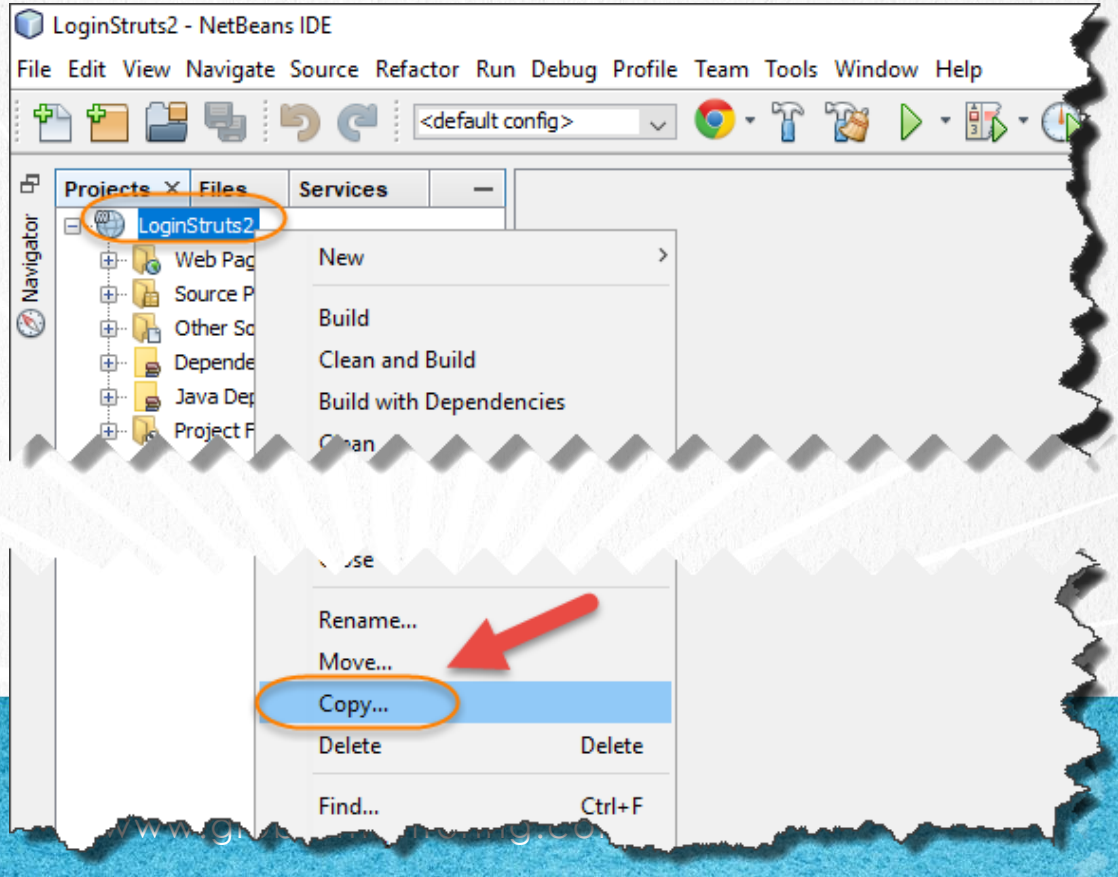
EXERCISE REQUIREMENT

We are going to start from the previous lab called LoginStruts2.

We are going to copy this project to work on this new lesson, in which we will add the concept of Results in Struts2.

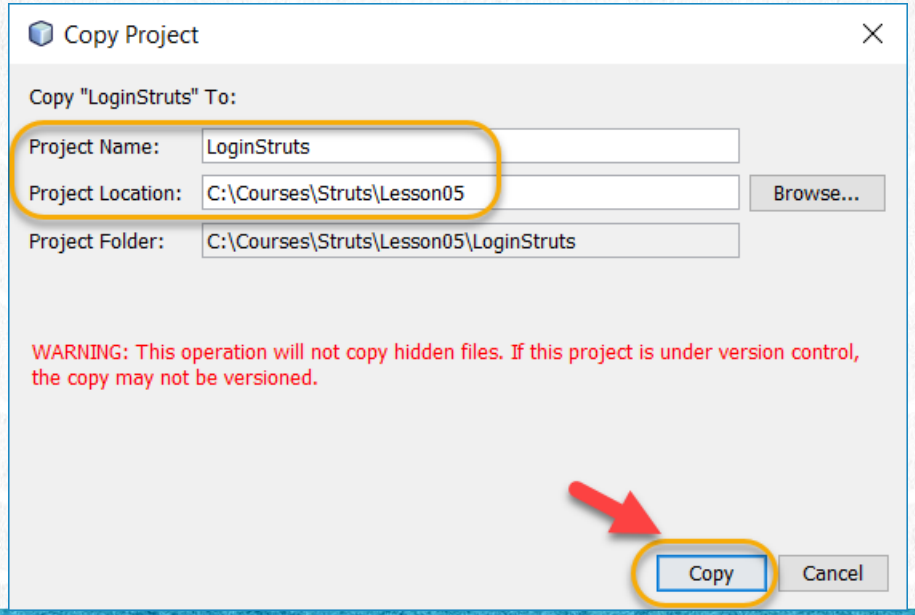
1. COPY THE PROJECT

- We created the new project using the copy function on the previous project:



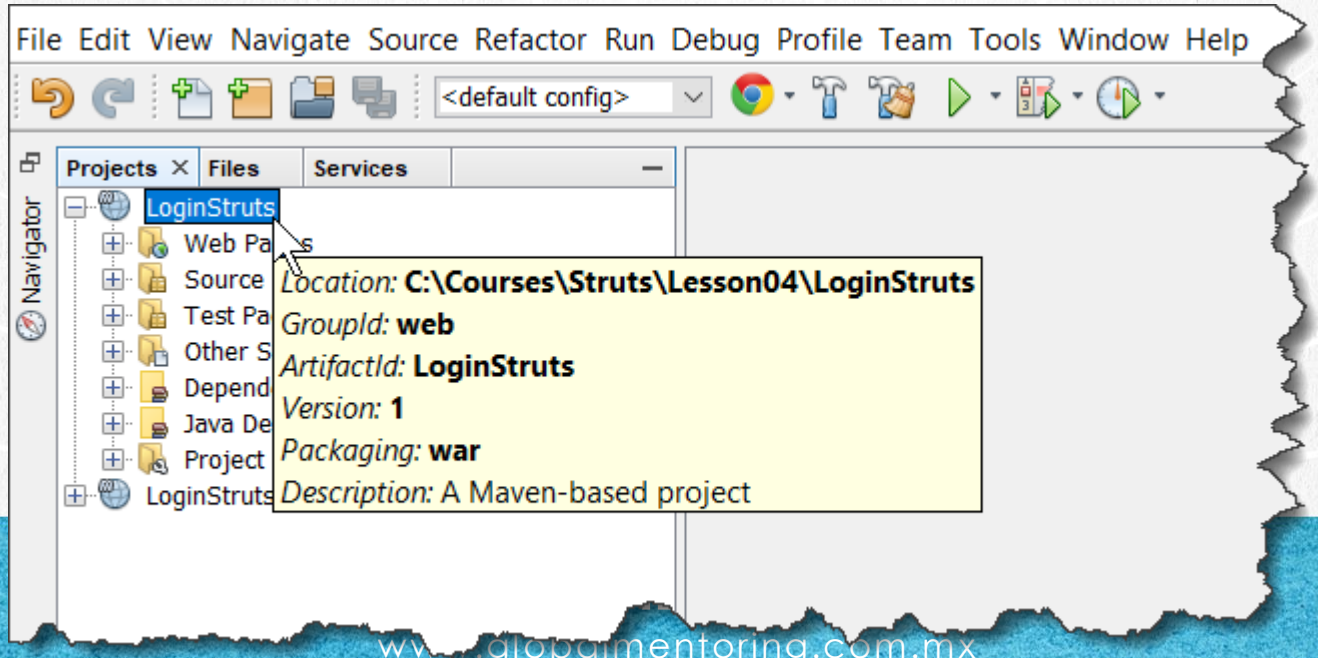
1. COPY THE PROJECT

- We write the following values as shown and click on copy. This will not change the name of the Maven project, so we'll leave it:



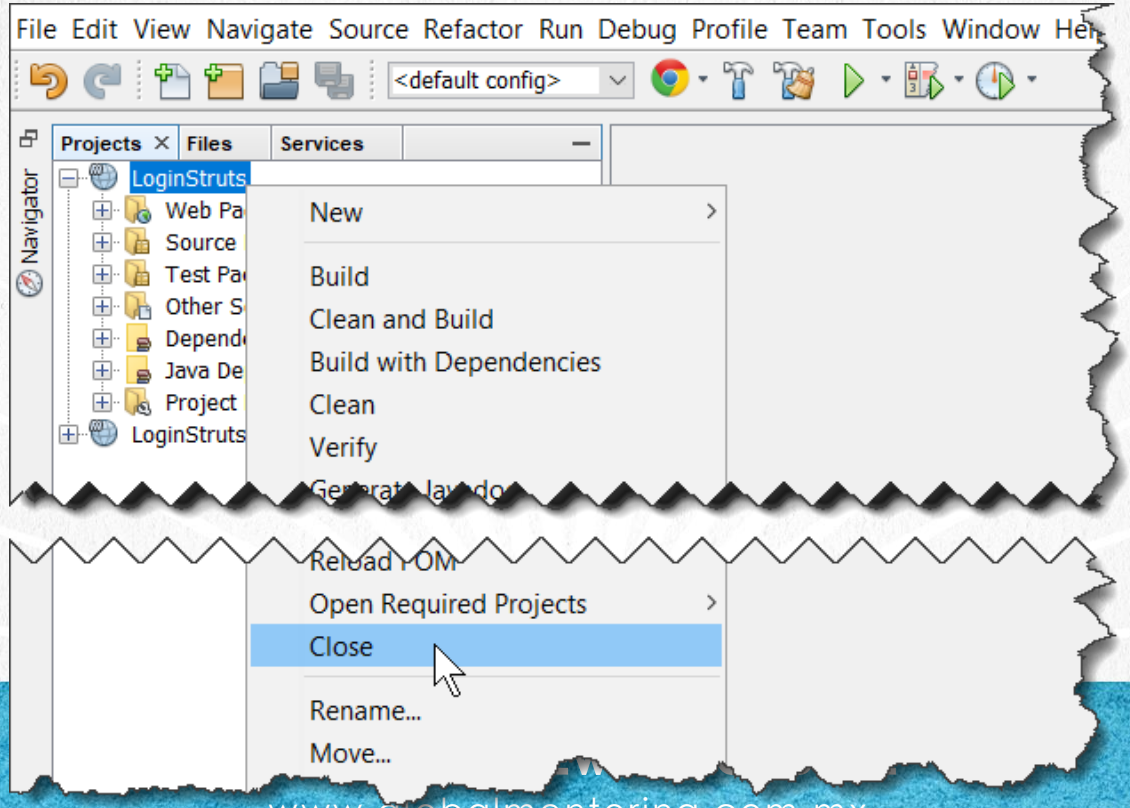
2. CLOSE THE PROJECT THAT WE NO LONGER USE

- We close the project that we no longer use, in this case to know which project we must close (Lesson04), we can position ourselves on the name of the project and it will give us more information, including the path where the file is located. This way we will know which project to close and which project to leave open:



2. CLOSE THE PROJECT THAT WE NO LONGER USE

- We close the project that we no longer use.



3. CREATE AN XML FILE

We add the **struts.xml** configuration file to define the results we are going to handle for each action we have defined.

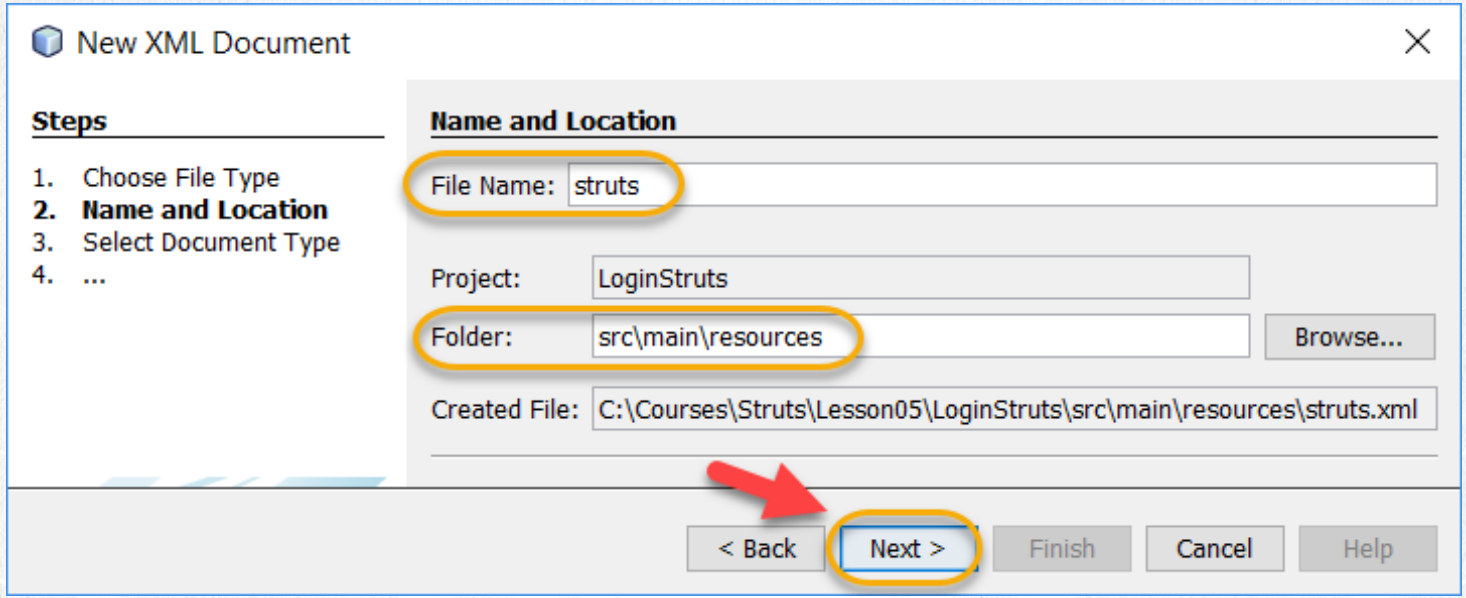
We will note that we can add actions that do not necessarily have an Action class associated with it, but only resend to a JSP (path = "login"). With this, what we achieve is that we continue using the MVC design pattern, even though it is a simple call to a JSP we are going through the framework without breaking the pattern and best practices of web programming.

On the other hand, we are defining an action (path = "validateUser") that can return two results, the first if the value of "admin" was provided in the user field, then the result is successful and the user is forwarded to the welcome.jsp view, if on the other hand, any other value is received in the user field, then the login.jsp page is displayed again.

Let's see how our struts.xml file is:

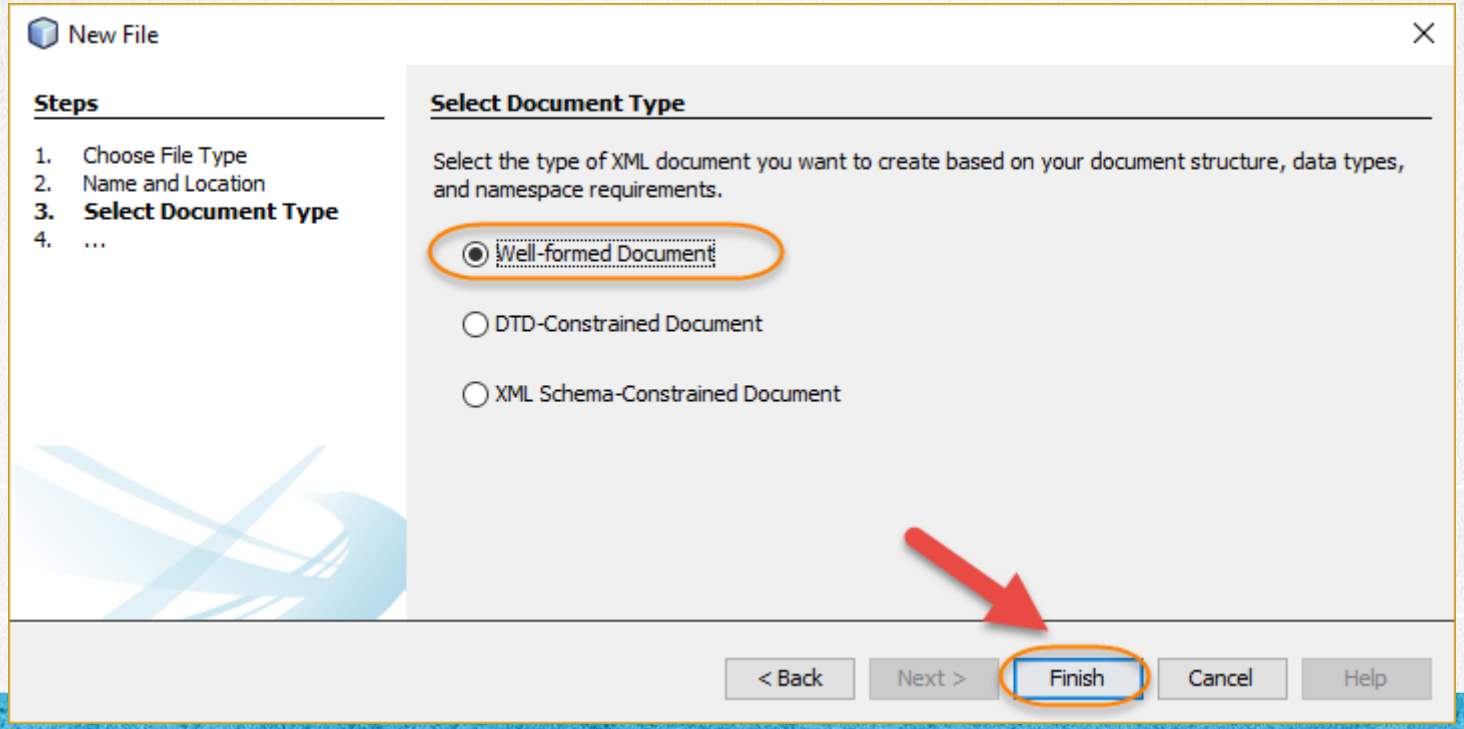
3 . CREATE AN XML FILE

- We create the struts.xml file, and deposit it in the route shown:



3 . CREATE AN XML FILE

- We select the option shown:



4. MODIFY THE FILE

struts.xml:

Click to download

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">
<struts>

    <constant name="struts.devMode" value="true" />

    <package name="default" extends="struts-default">
        <!--path /login without specifying a class of type Action, only a result-->
        <action name="login">
            <result>/WEB-INF/content/login.jsp</result>
        </action>
        <!--Action with two possible results, one of a successful type,
         and another a redirect to the newly defined login action-->
        <action name="validateUser" class="web.actions.LoginAction">
            <result name="success">/WEB-INF/content/welcome.jsp</result>
            <result name="input" type="redirectAction">login</result>
        </action>
    </package>

</struts>
```

5. MODIFY THE ACTION CLASS

We modify the LoginAction class so that we can return different results depending on the value received in the "user" property.

We will also use a different syntax for handling messages, so it will not be necessary to map the attributes for each of the messages, but we can access them directly in our JSP with the syntax:

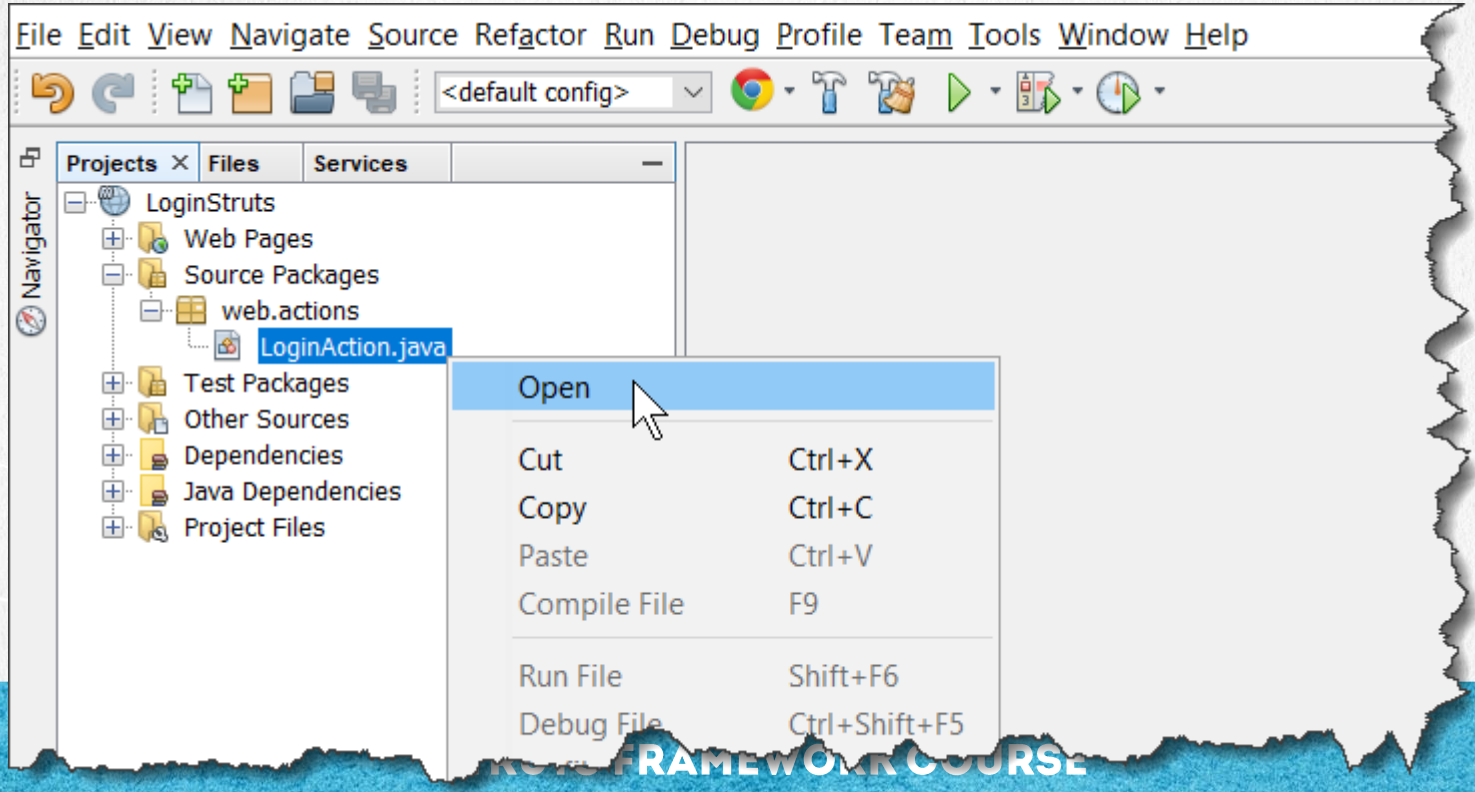
<s:text name = "propertyName" />

The execute method is the one that will validate if the user provided is equal to "admin" or not. For the moment it will be programmed in hard code (fixed string of "admin"), however at the end of the course you can easily connect to any database to do this validation dynamically against a database table.

Depending on the value provided, the result will return "success" in case of correct value, or "input" in case of incorrect return to the user capture form and password. We are only validating the user field for the moment to simplify the exercise.

5. MODIFY THE ACTION CLASS

- We open the LoginAction.java class to modify it:



5. MODIFY THE CODE

LoginAction.java:

Click to download

```
package web.actions;

import com.opensymphony.xwork2.ActionSupport;
import org.apache.logging.log4j.*;

public class LoginAction extends ActionSupport {

    private String user;
    private String password;

    Logger log = LogManager.getLogger(LoginAction.class);

    @Override
    public String execute() {
        //If it is valid user we show the welcome page.jsp
        if ("admin".equals(this.user)) {
            return SUCCESS;
        } else {
            //If it is user NOT valid, we return to the login
            return INPUT;
        }
    }
```

5. MODIFY THE CODE

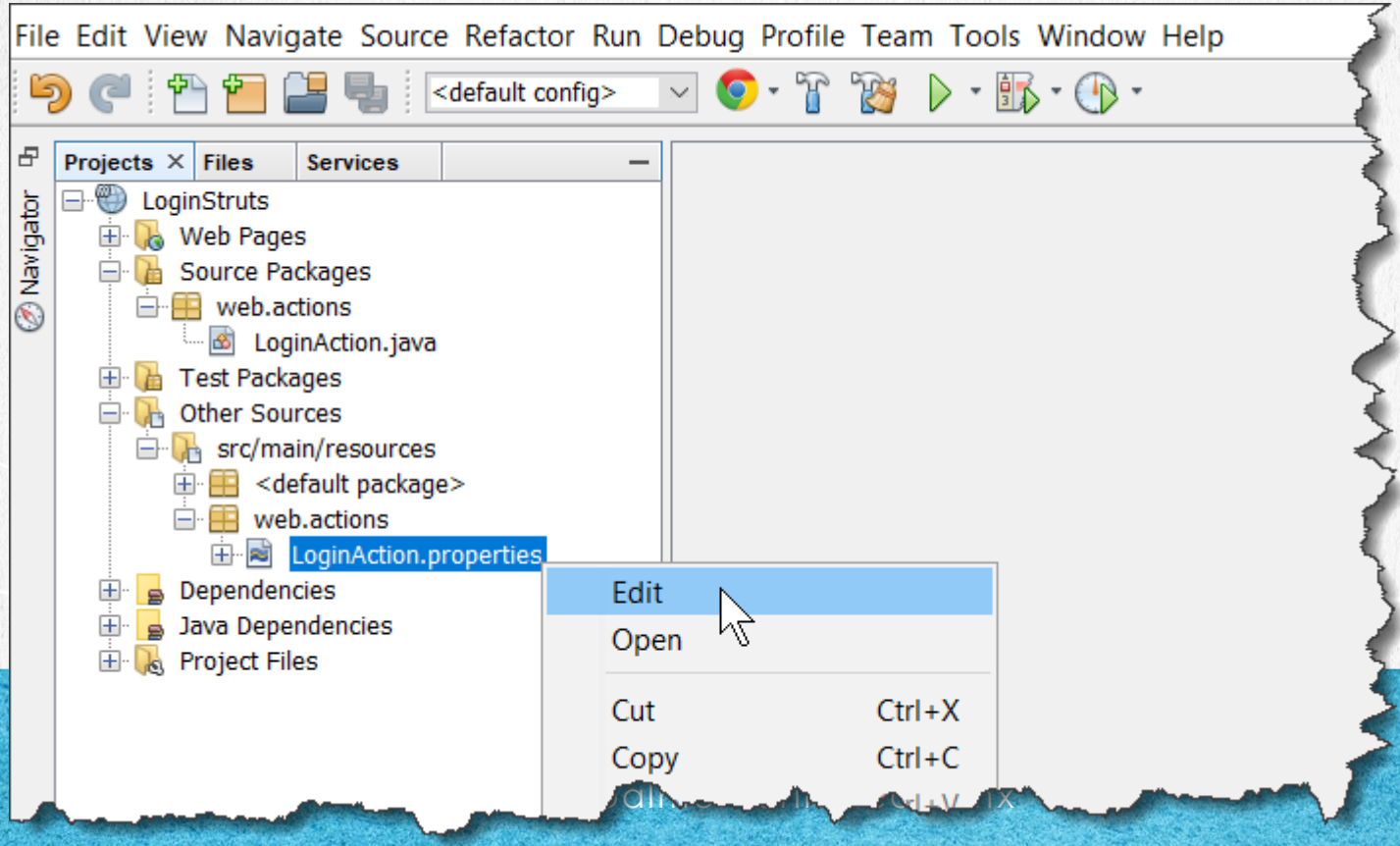
LoginAction.java:

Click to download

```
    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

}
```

6. MODIFY THE PROPERTIES CLASS

- Open the LoginAction.properties file to modify it :



PASO 6. MODIFICAMOS EL CÓDIGO

LoginAction.properties:

Clic para ver el archivo

```
form.user: User
form.password: Password
form.button: Send
form.title: Login
welcome.title: Welcome
welcome.message: Correct User
```

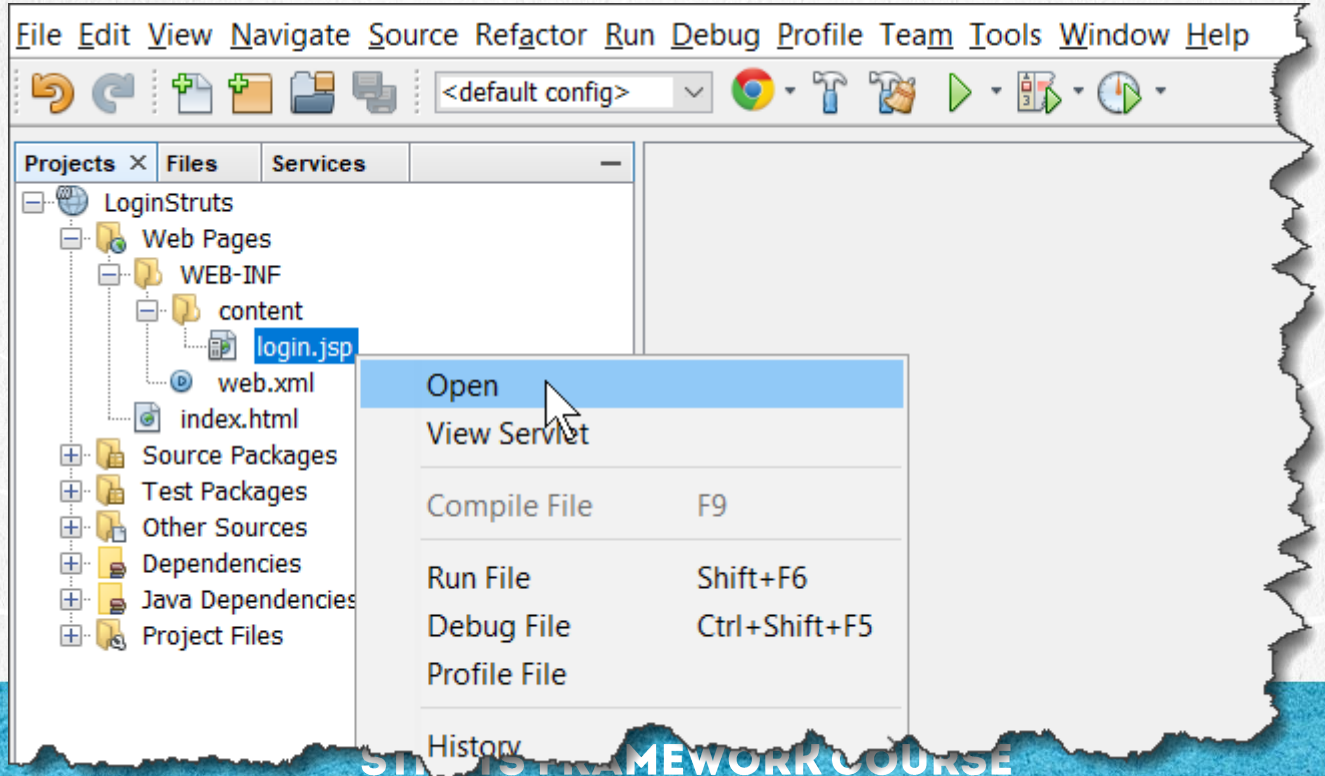
7. MODIFY THE JSP

Now we modify the JSP of login.jsp to adapt it to the new changes in the messages that we are going to use, as well as the flow that the action of LoginAction.java will be sent.

In this case we will not use the configuration by convention, we will make use of the configuration that we have added in the struts.xml file, however it is possible to use the configuration by convention to simplify the configuration and combining it with Struts annotations to eliminate the struts.xml file completely. This we will do later.

7. MODIFY THE JSP

- We open the login.jsp file:



7. MODIFY THE CODE

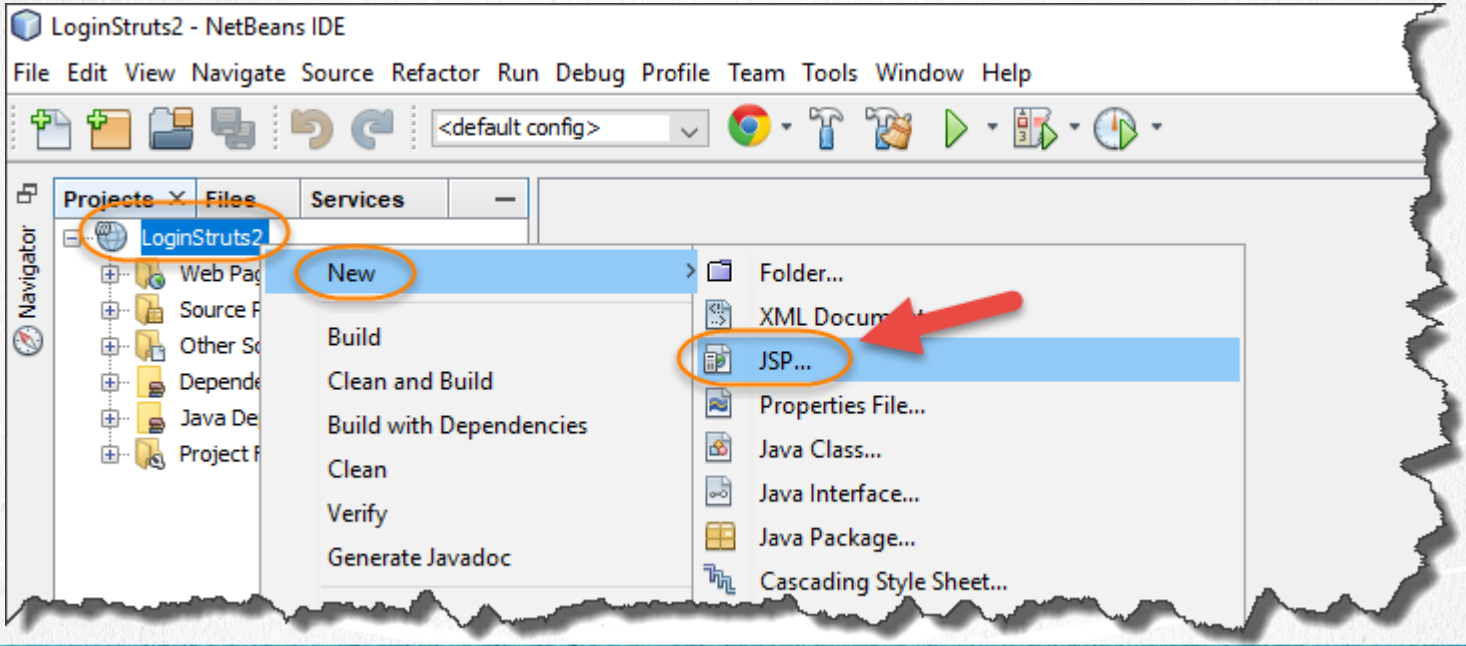
login.jsp:

Click to download

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib  prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
    <head>
        <title><s:text name="form.title" /></title>
    </head>
    <body>
        <%-- The url is of the form is: /validateUser --%>
        <h1><s:text name="form.title" /></h1>
        <s:form action="validateUser">
            <s:textfield key="form.user" name="user" />
            <s:password key="form.password" name="password" />
            <s:submit key="form.button" name="submit" />
        </s:form>
    </body>
</html>
```

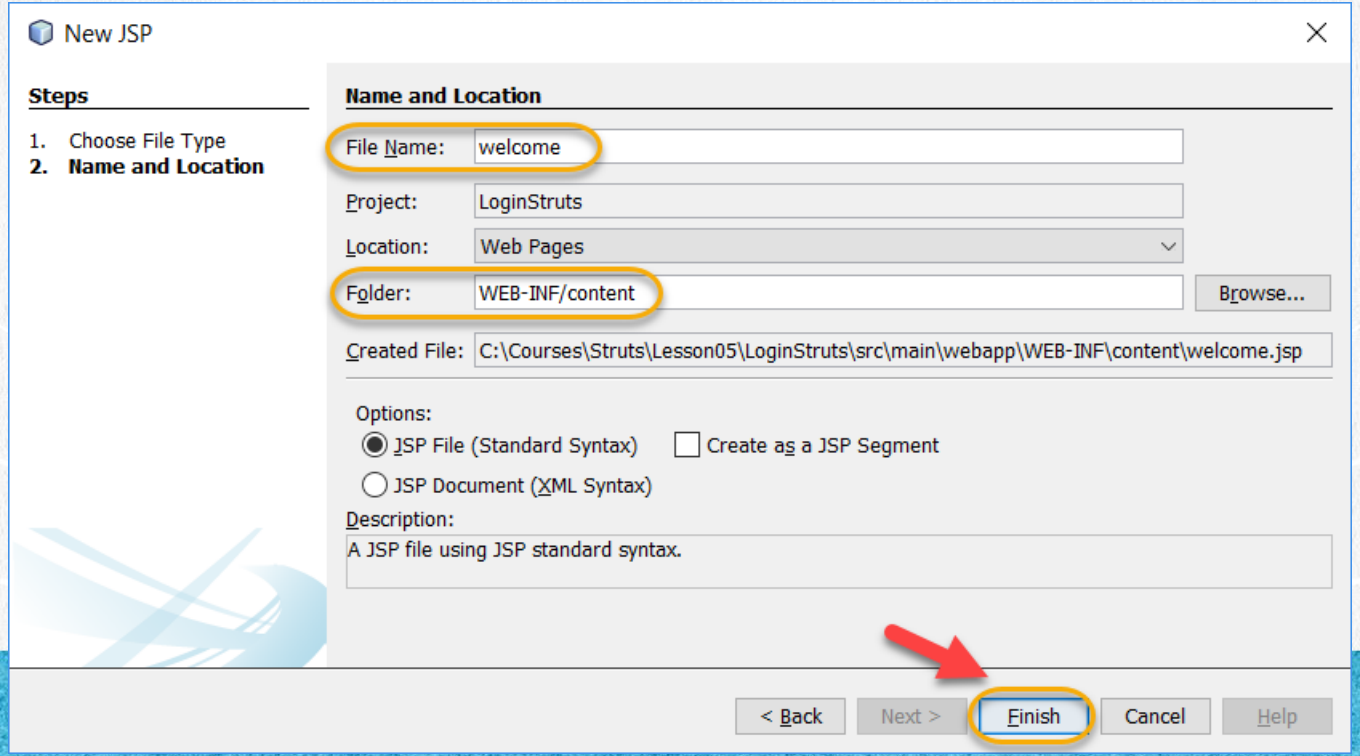
8. CREATE A NEW JSP

We created the welcome.jsp file:



8. CREATE A NEW JSP

We created the welcome.jsp file



9. MODIFY THE CODE

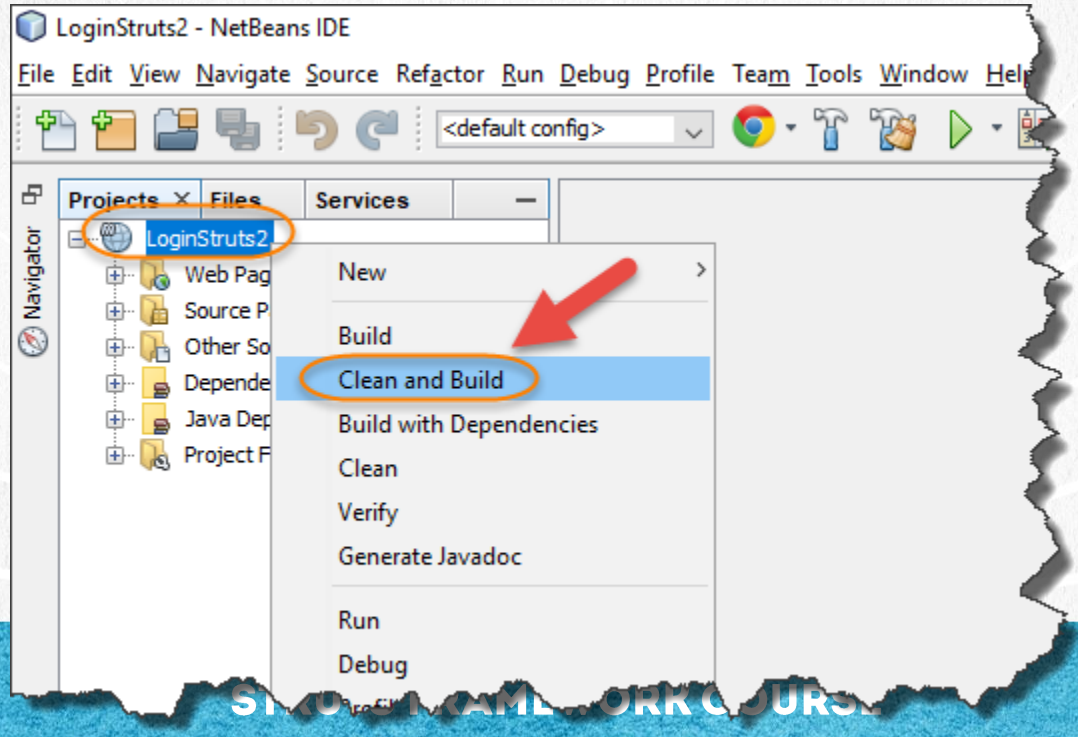
welcome.jsp:

Click to download

```
<%@ page contentType="text/html; charset=UTF-8" %>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
    <head>
        <title><s:text name="welcome.title" /></title>
    </head>
    <body>
        <h1><s:text name="welcome.title" /></h1>
        <h2>
            <s:text name="welcome.message" />: <s:property value="user"/>
        </h2>
    </body>
</html>
```

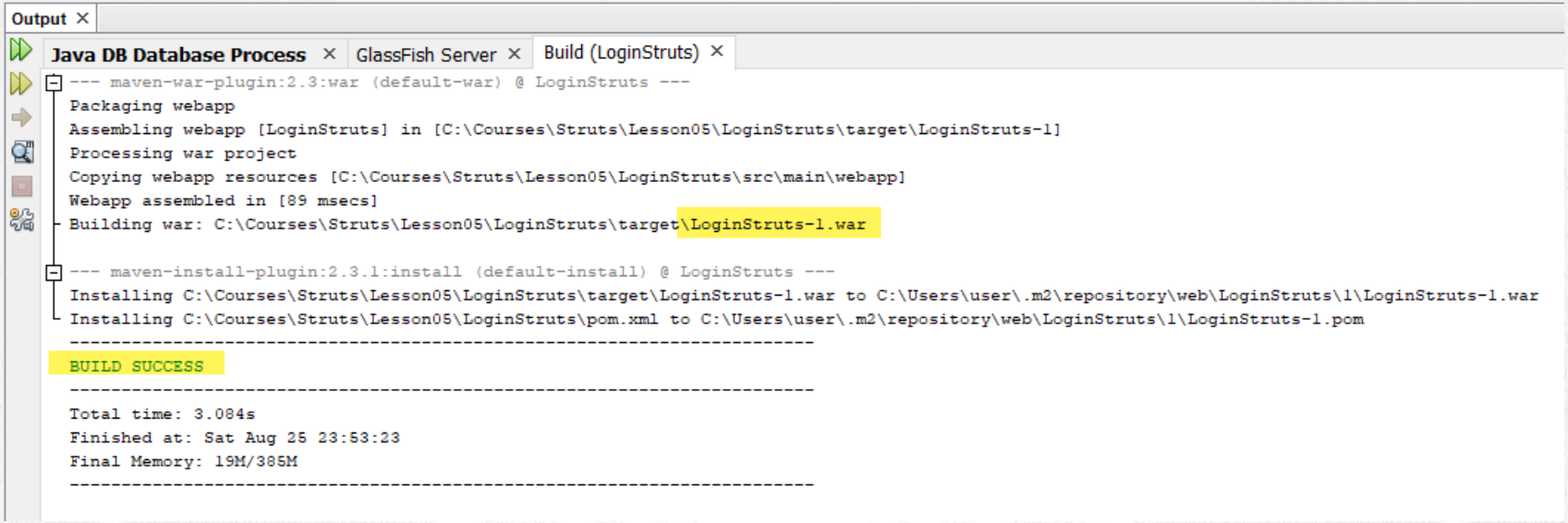
10. EXECUTE CLEAN & BUILD

- First stop Glassfish if it is running, and later execute the Clean & Build option as show:



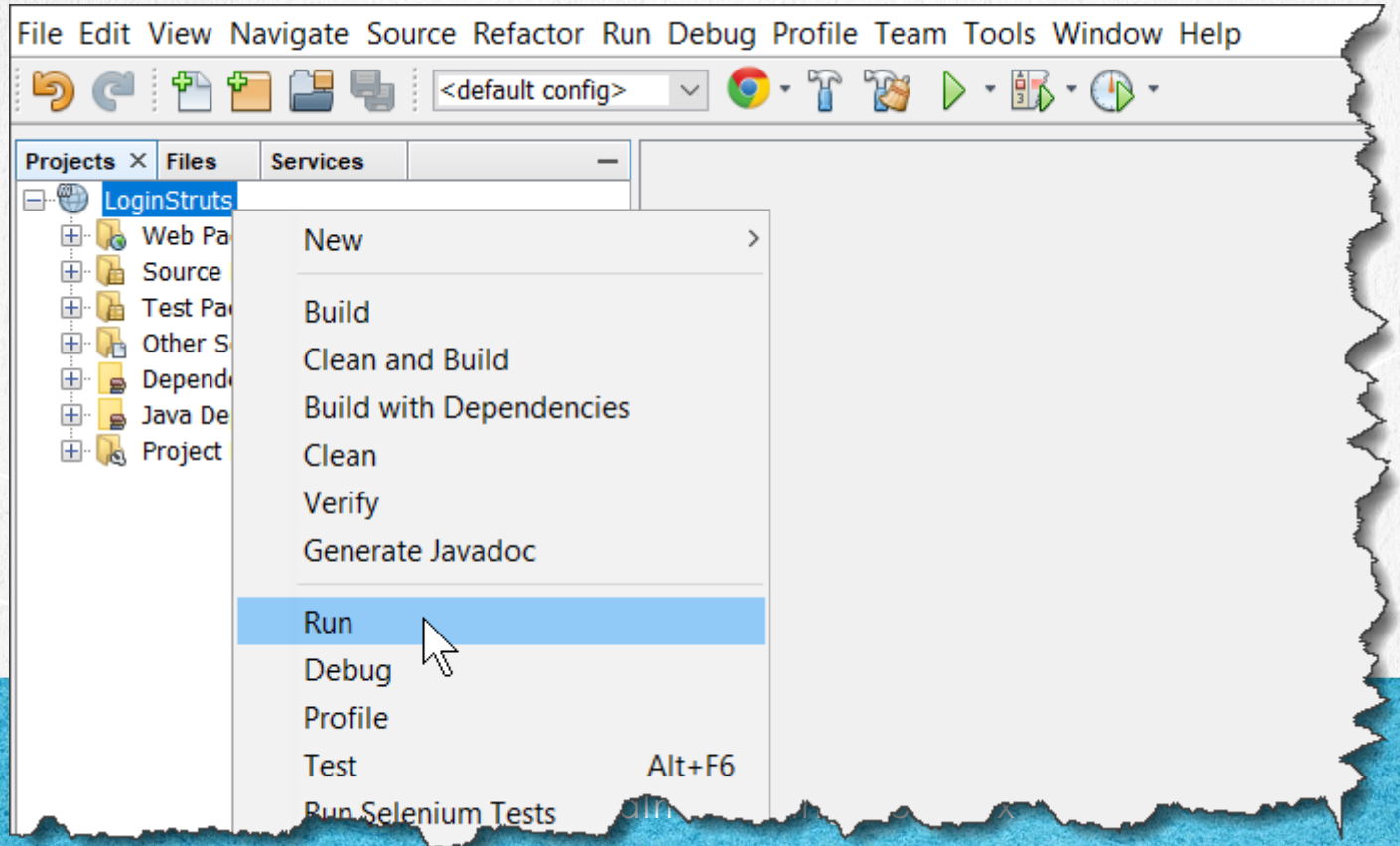
10. EXECUTE CLEAN & BUILD

- We should observe a result similar to the following:



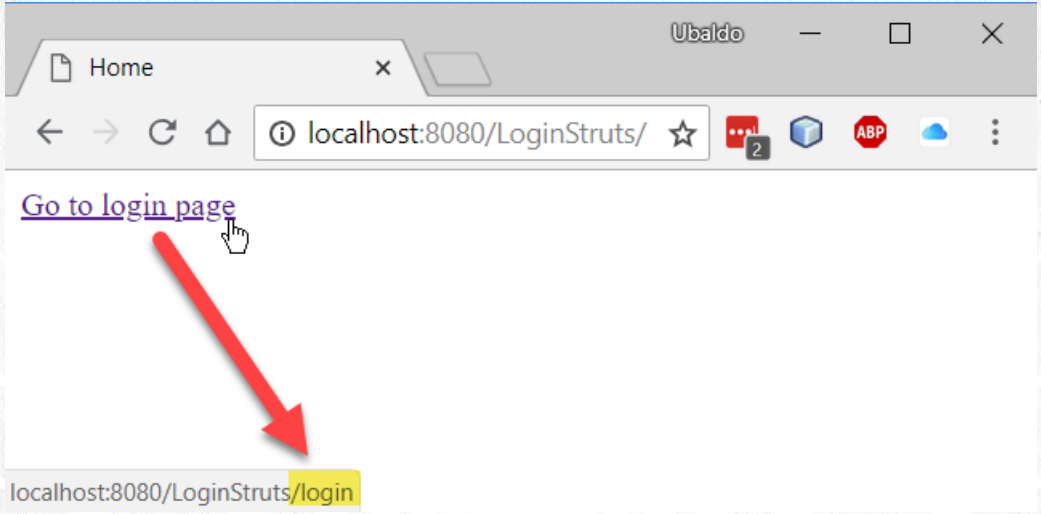
11. EXECUTE THE APPLICATION

- We execute the LoginStruts application as follows :



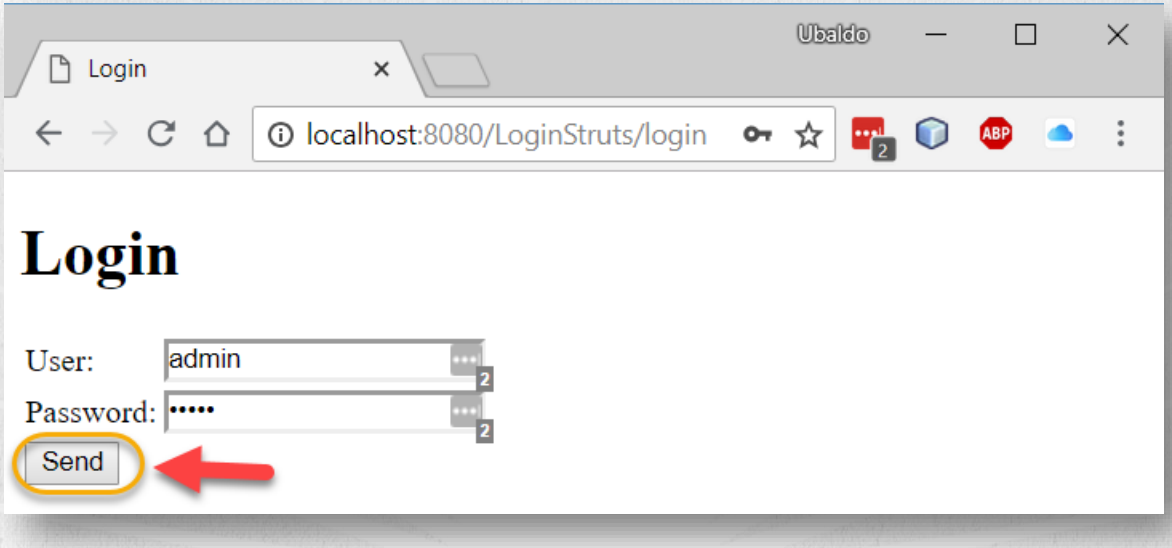
11. EXECUTE THE APPLICATION

- Execute the application as follows:



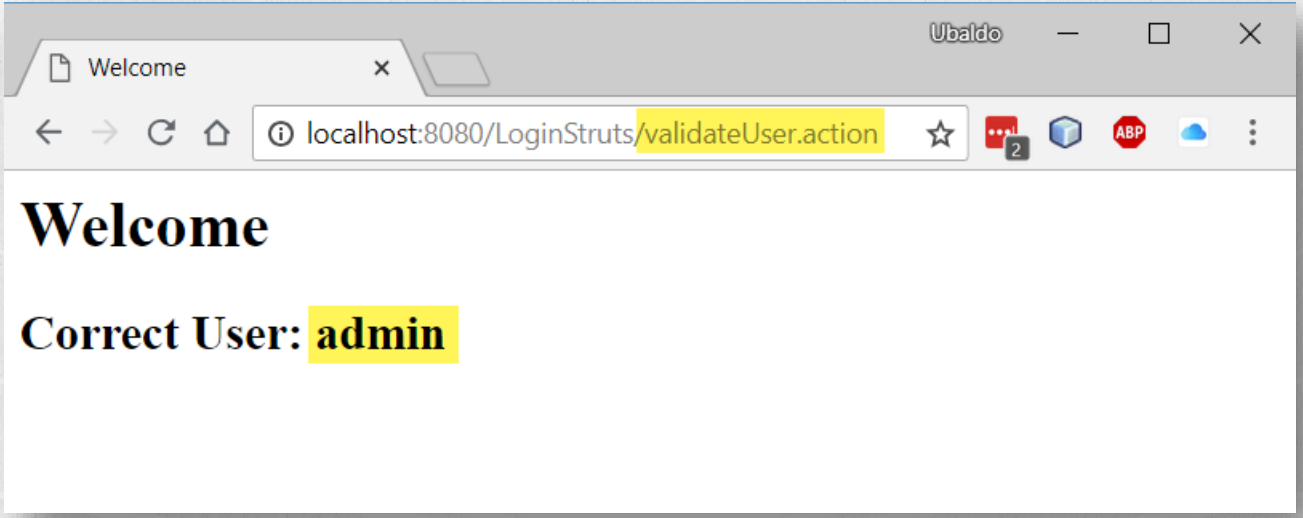
11. EXECUTE THE APPLICATION

- We look at the form, and if we provide the value of "admin" in the user field it will direct us to the path of / validateUser and the result will be "success" showing us as a result the welcome view.jsp:



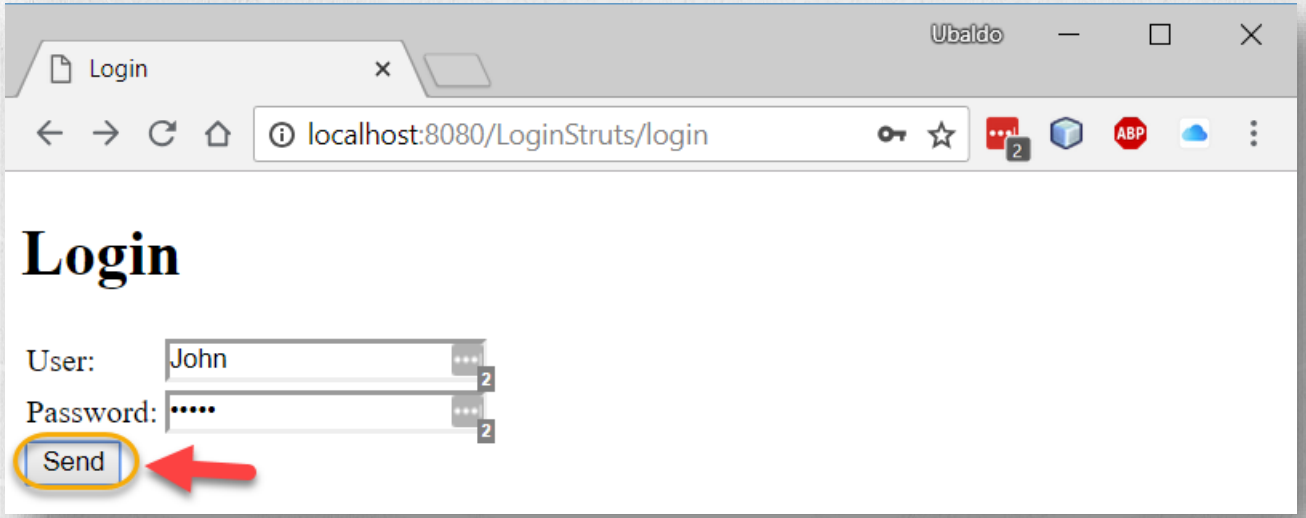
11. EXECUTE THE APPLICATION

- We observe the value provided by the user, and we see that the action that was executed is the path of: /validateUser, showing us the view of welcome.jsp



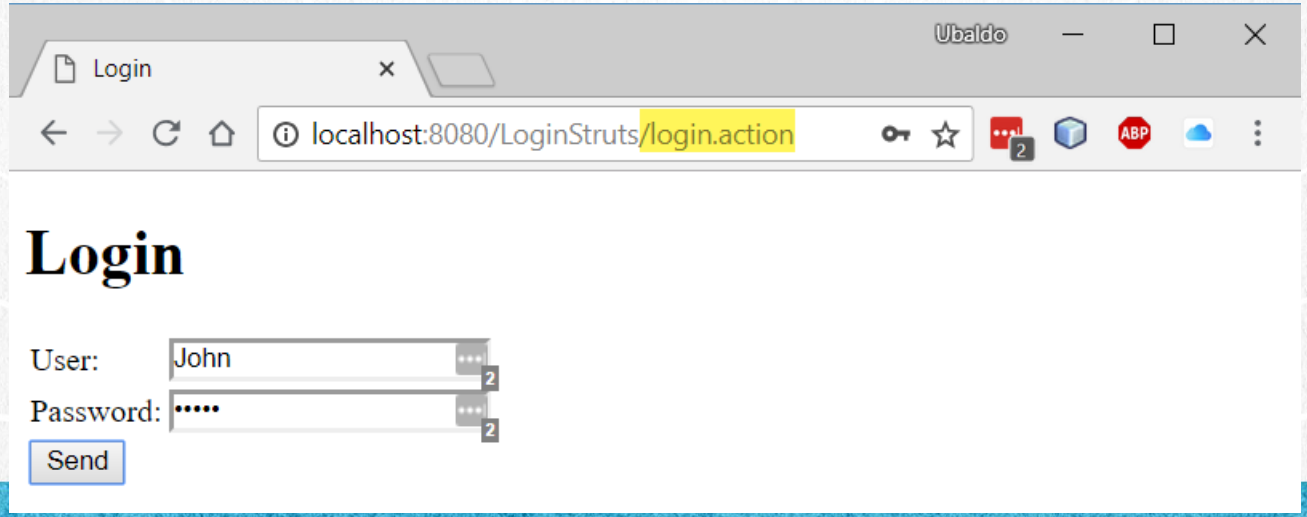
11. EXECUTE THE APPLICATION

- We observe the form, and if we provide the value other than "admin" in the user field, it will direct us to the path of /validateUser and the result will be "input" showing us the result of the login.jsp view again:



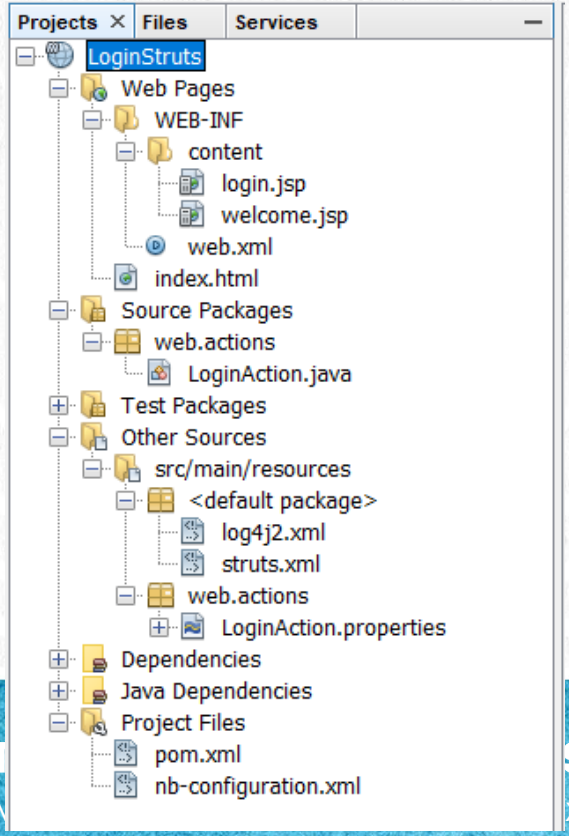
11. EXECUTE THE APPLICATION

•We see that the action that was executed is the path of: validateUser, and because the user value was not equal to "admin", then it returned as a result the "input" string, showing us again the login.jsp view. To do this, we configured a redirect in the struts.xml file to the "login" path, and that again showed the JSP of login.jsp, and modifying the URL to show the path of login.action, instead of vadlidarUsuario. action which is the path that would have been shown if we had not specified the redirect to the "login" action:



FINAL STRUCTURE OF THE PROJECT

At the end of the exercise the structure should be as follows.



FINAL RECOMMENDATIONS

If for some reason the exercise fails, several things can be done to correct it:

Stop the Glassfish server
Make a Clean & Build project to have the most recent version compiled
Restart the project (deploy the project to the server again)

If the above does not work, you can try loading the resolved project which is 100% functional and rule out configuration problems in your environment or any other code error.

Experiencia y Conocimiento para tu vida

EXERCISE CONCLUSION

With this exercise we put into practice the handling of Results in Struts 2.

There are several types of results in Struts 2, and you can see more information in the following links:

https://struts.apache.org/core-developers/result-types.html

https://struts.apache.org/core-developers/result-configuration.html

With this exercise we are ready to add more features, such as validation issues, error handling, among several other concepts that we will see later.

ONLINE COURSE

STRUTS 2 FRAMEWORK

By: Eng. Ubaldo Acosta