

# HƯỚNG DẪN TẠO PROJECT LOGIN STRUTS2.

## I. Giới thiệu và chuẩn bị

Trong file word này, sẽ hướng dẫn các bạn viết một ứng dụng struts 2 về chức năng đăng nhập (login). Trước hết, chúng ta cần phải chuẩn bị các thứ sau:

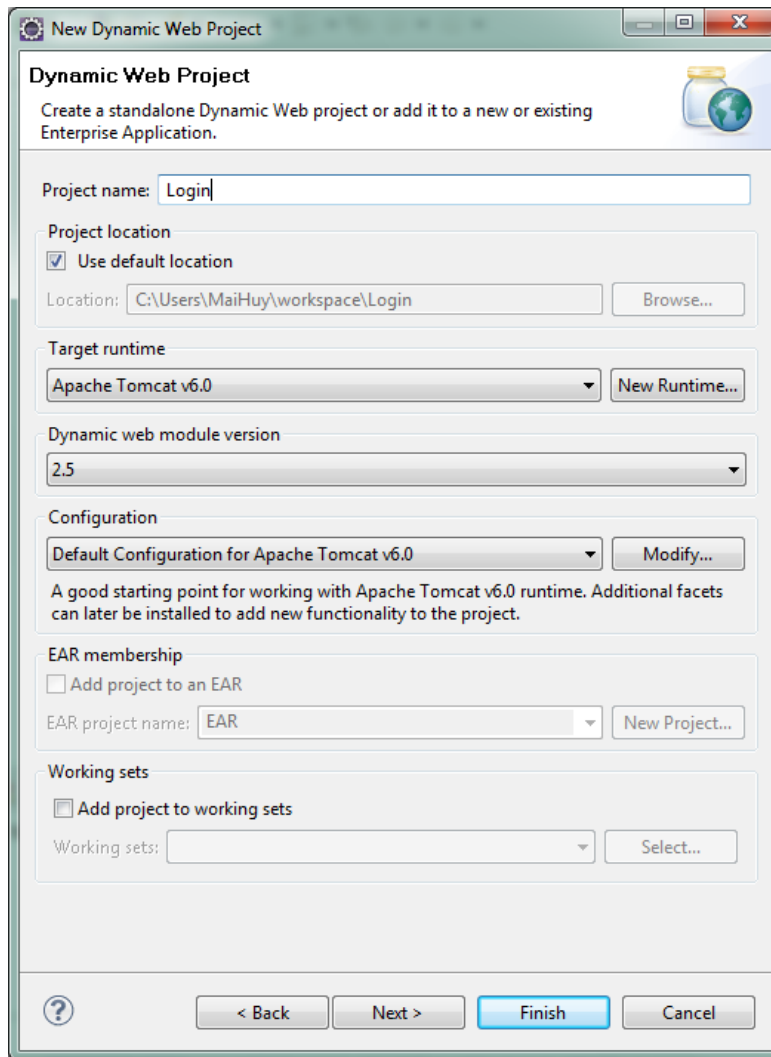
- Phiên bản JDK 1.5 trở lên
- Apache Tomcat 6.x
- Eclipse 3.2 trở lên
- Các thư viện của Struts2:
  - o commons-logging-1.0.4.jar
  - o freemarker-2.3.16.jar
  - o ognl-3.0.jar
  - o struts2-core-2.2.1.jar
  - o xwork-core-2.2.1.jar

Lưu ý rằng tùy theo phiên bản hiện hành của Struts2, số phiên bản của các tập tin jar ở trên có thể thay đổi

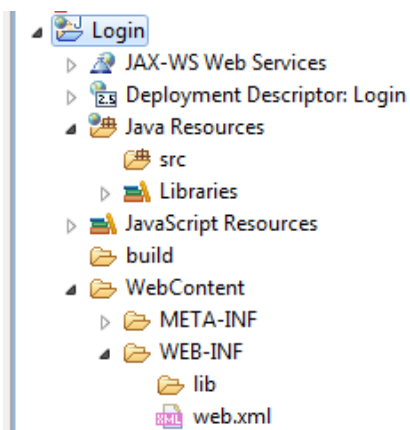
## II. Getting Started

### 1. Getting Started

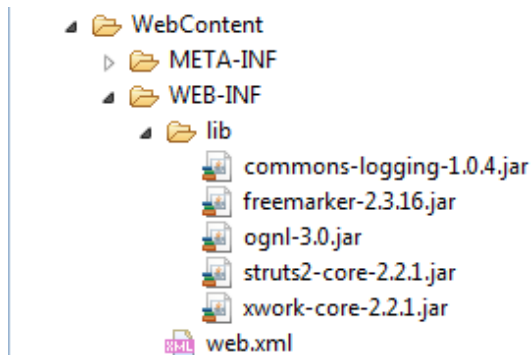
- Mở chương trình Eclipse, và chọn File -> New -> Project and select **Dynamic Web Project**



- Sau đó chúng ta chọn Finish, và được một project có cấu trúc như sau:



- Tiếp theo, chúng ta copy các tập tin JAR sau và WebContent→WEB-INF→lib (tạo folder lib nếu không có).



## 2. Cấu hình web.xml

- web.xml là một tập tin định dạng XML, dùng để cấu hình Project. Chúng ta sẽ định nghĩa Filter là `org.apache.struts2.dispatcher.FilterDispatcher` trong web.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">

  <display-name>Login</display-name>

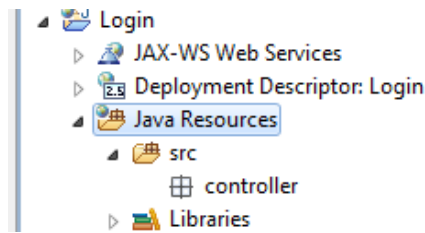
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.FilterDispatcher
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>

  </filter-mapping>
  <welcome-file-list>
    <welcome-file>Login.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

## 3. Action class

- Chúng ta cần một class Action để làm nhiệm vụ kiểm tra xác thực username và password của user.
- Trước hết, chúng ta tạo một package là **controller** trong folder **src** để chứa class Action này: click phải folder src → new → package → tạo package xong refresh Project Explorer ta được như sau:



- Tiếp theo, chúng ta tạo một class là LoginAction trong package controller: click phải package controller → New → Class → đặt tên là LoginAction → Finish
- Class LoginAction ta có đoạn mã sau:

```
package controller;

public class LoginAction {
    private String username;
    private String password;

    public String execute() {
        if (this.username.equals("admin")
            && this.password.equals("admin123")) {
            return "success";
        } else {
            return "error";
        }
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

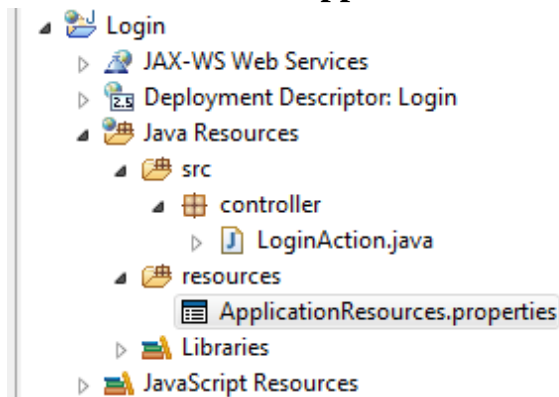
    public String getPassword() {
        return password;
    }

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

- Chú ý:
  - o Trong class này có 2 thuộc tính là **username** và **password**, sẽ dùng để chứa giá trị nhập từ form và class này cũng có phương thức **execute()** dùng để chứng thực user. Trong ví dụ này, chúng ta mặc định username là *admin* và password là *admin123*.
  - o Phương thức **execute()** trả về kiểu giá trị là String, dùng để xác định các trang để gửi về người dùng.

#### 4. ResourceBundle

- ResourceBundle rất hữu dụng trong việc chứa các **nội dung tĩnh** trong một file. Hầu hết các ứng dụng đều định nghĩa ResourceBundle trong file **\*.properties**, dùng để chứa nội dung các message, ví dụ trong trường hợp này là *“Username hoặc Password không đúng”*.
- Chúng ta sẽ tạo và định nghĩa file **ApplicationResources.properties**. Trước tiên, chúng ta tạo một source folder là **resources**: Click phải Project → New → Sources Folder → đặt tên là resources → Ok.
- Tiếp theo, click phải resources → New → Other → General → File → Next → đặt tên là **ApplicationResources.properties** → Finish.



- Nội dung ApplicationResources.properties là như sau:

```
label.username= Username  
label.password= Password  
label.login= Login
```

#### 5. JSP

- Chúng ta sẽ tạo 2 file JSP để hiển thị cho user. **Login.jsp** sẽ là trang mặc định của ứng dụng, gồm một form login đơn giản với 2 trường là username và password. Khi đăng nhập thành công, user sẽ được load qua trang **Welcome.jsp** với lời chào mừng.
- Tạo 2 file JSP là Login.jsp và Welcome.jsp trong folder WebContent, click phải WebContent → New → JSP File, với nội dung như sau:  
Login.jsp

```
<%@ page contentType="text/html; charset=UTF-8"%>  
<%@ taglib prefix="s" uri="/struts-tags"%>  
<html>  
<head>  
  
<title>Struts 2 - Login Application</title>  
</head>  
  
<body>  
<h2>Struts 2 - Login Application</h2>  
<s:actionerror />
```

```

<s:form action="login.action" method="post">

    <s:textfield name="username" key="label.username"
size="20" />
    <s:password name="password" key="label.password"
size="20" />

    <s:submit method="execute" key="label.login"
align="center" />
</s:form>
</body>
</html>

```

## Welcome.jsp

```

<%@ page contentType="text/html; charset=UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<html>

<head>
<title>Welcome</title>
</head>

<body>
    <h2>Hello, <s:property value="username" />...!</h2>

</body>
</html>

```

- Chú ý: ta sử dụng tag <s:> trong struts2 để hiển thị textbox và label. Struts2 cung cấp một thư viện tag khá mạnh mẽ và phong phú để hiển thị giao diện.

### 6. Định nghĩa struts.xml

- Struts2 sẽ đọc cấu hình và định nghĩa các class từ **struts.xml**. Do đó, chúng ta sẽ định nghĩa struts.xml trong resource folder. Tạo file struts.xml trong folder resources : Click phải resources chọn New → Other → XML → XML file → Đặt tên là struts.xml → Finish.
- File struts.xml có nội dung như sau:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration
    2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

    <constant name="struts.enable.DynamicMethodInvocation"
        value="false" />
    <constant name="struts.devMode" value="false" />
    <constant name="struts.custom.i18n.resources"
        value="ApplicationResources" />

```

```

<package name="default" extends="struts-default"
        namespace="/">

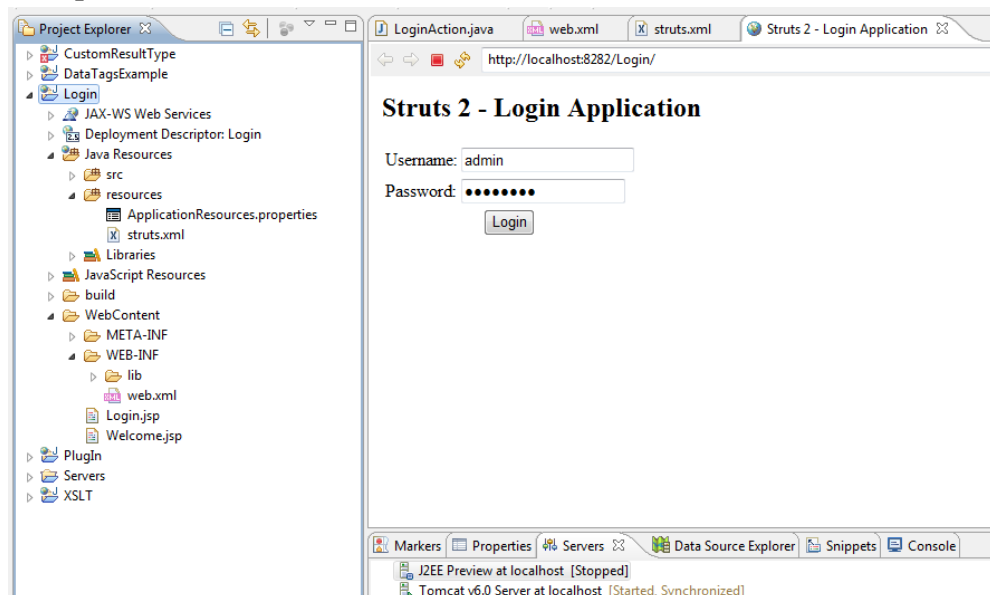
    <action name="login"
            class="controller.LoginAction">
        <result name="success">Welcome.jsp</result>
        <result name="error">Login.jsp</result>
    </action>
</package>
</struts>

```

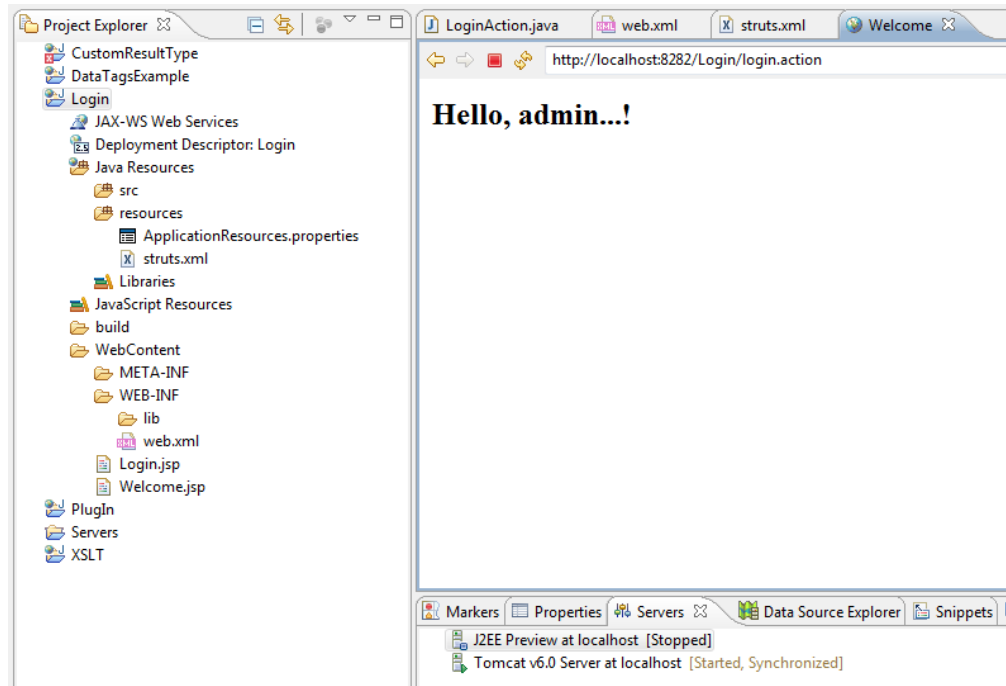
- Chú ý:
  - Trong file trên, chúng ta đã định nghĩa Login action. Có 2 kết quả được là “success” và “error”, dựa vào kết quả trả về của hàm execute() trong LoginAction.
  - Chúng ta cũng khai báo một hằng số có tên là struts.custom.i18n.resources. Hằng số này có liên kết một file resource bundle mà chúng ta đã tạo bước 4.
  - Class LoginAction có chứa phương thức execute(), đây là phương thức mặc định khi gọi class này.

## 7. Hoàn tất

- Cơ bản chúng ta đã hoàn thành project, bạn có thể chạy và kiểm tra lại thử.
- Để chạy project, click phải tên Project chọn select Run as → Run on Server.
- Kết quả ta được như sau:



- Sau khi đăng nhập Username là admin và password là admin123 ta sẽ chuyển tới trang Welcome.jsp



### III. Nâng cao

- Trong Project trên, vẫn còn nhược điểm, khi chúng ta đăng nhập thất bại, giao diện không có gì thay đổi để nhắc nhở user đăng nhập sai.
- Để thêm chức năng này, chúng ta cần thêm một message báo lỗi trong file ResourceBundle. Lúc này, nội dung file ApplicationResources.properties có nội dung như sau

```
label.username= Username
label.password= Password
label.login= Login
error.login= Invalid Username/Password. Please try again.
```

- Ngoài ra, chúng ta cần chỉnh cho LoginAction xử lý xuất hiện message báo lỗi khi đăng nhập thất bại. Lúc này, LoginAction có nội dung như sau:

```
package controller;

import com.opensymphony.xwork2.ActionSupport;

public class LoginAction extends ActionSupport {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private String username;
    private String password;

    public String execute() {

        if (this.username.equals("admin")
            && this.password.equals("admin123")) {
```



```

        return "success";
    } else {
        addActionError(getText("error.login"));
        return "error";
    }
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

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

```

- Cuối cùng, chúng ta chạy lại project và kiểm tra thử. Khi chúng ta đăng nhập thất bại, thì trang Login.jsp sẽ hiển thị thông báo lỗi

