

Chương trình Struts 2 đầu tiên

Trong chương này, chúng ta sẽ tìm hiểu cách tạo một ví dụ khác cho Struts 2 mà không sử dụng IDE. Trước hết, chúng ta cần tạo 4 thành phần cho bất cứ Struts 2 project nào.

Action: Tạo một lớp Action chứa trình logic và điều khiển tương tác giữa người dùng, model và view.

Interceptor: Tạo Interceptor nếu cần thiết, hoặc sử dụng bất cứ Interceptor nào đã tồn tại. Đây là một phần của Controller.

View: Tạo một JSP để tương tác với người dùng để nhận input và để biểu diễn các thông điệp cuối cùng.

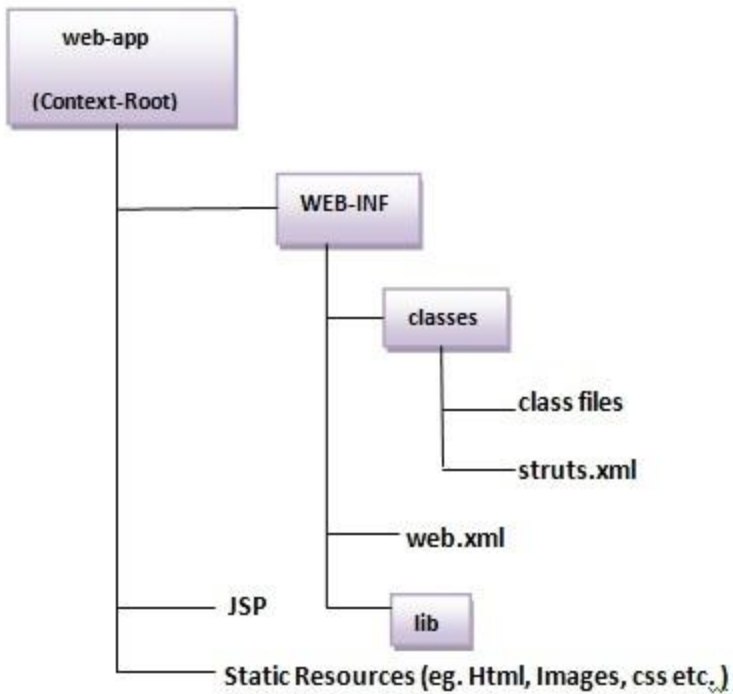
Các file cấu hình: Tạo các file cấu hình cho Action, View và Controller. Các file này là struts.xml, web.xml và struts.properties.

Dưới đây là các bước chi tiết để tạo ứng dụng Struts 2:

- Tạo cấu trúc thư mục
- Tạo index.jsp
- Cung cấp entry của Controller trong web.xml file
- Tạo lớp Action (ví dụ Product.java)
- Ánh xạ request với Action trong struts.xml file và định nghĩa các thành phần view
- Tải jar file
- Bắt đầu Server và triển khai ứng dụng

Bước 1: Tạo cấu trúc thư mục

Cấu trúc thư mục của Struts 2 là giống như Servlet/JSP. Ở đây, struts.xml file phải được đặt trong class folder.



Bước 2: Tạo input page (index.jsp)

JSP page này tạo một form bởi sử dụng các thẻ UI trong Struts. Để sử dụng Struts UI tag, bạn cần xác định uri là /struts-tags. Ở đây, chúng ta đã sử dụng s:form để tạo một form, s:textfield để tạo một trường text, và s:submit để tạo một nút Submit.

index.jsp

```
<%@ taglib uri="/struts-tags" prefix="s" %>
<s:form action="product">
<s:textfield name="id" label="Product Id"></s:textfield>
<s:textfield name="name" label="Product Name"></s:textfield>
<s:textfield name="price" label="Product Price"></s:textfield>
<s:submit value="save"></s:submit>
</s:form>
```

Bước 3: Cung cấp entry của Controller trong web.xml file

Trong Struts 2, lớp StrutsPrepareAndExecuteFilter làm việc như là Controller. Filter này được cung cấp ngầm định bởi Struts Framework.

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

Bước 4: Tạo lớp Action

Đây là lớp Bean đơn giản. Trong Struts 2, Action là POJO (viết tắt của Plain Old Java Object). Nó có một phương thức có tên là execute được triệu hồi bởi Struts Framework theo mặc định.

product.java

```
package com.vietjack;

public class Product {
  private int id;
  private String name;
  private float price;
  public int getId() {
    return id;
  }
  public void setId(int id) {
    this.id = id;
  }
  public String getName() {
```

```
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public float getPrice() {
        return price;
    }
    public void setPrice(float price) {
        this.price = price;
    }

    public String execute(){
        return "success";
    }
}
```

Bước 5: Ánh xạ request trong struts.xml file và định nghĩa các thành phần view

Đây là file quan trọng, tại đây Struts Framework lấy thông tin về Action và quyết định kết quả để được triệu hồi. Ở đây, chúng ta đã sử dụng nhiều phần tử như struts, package, action và result.

Phần tử **struts** là phần tử gốc của file này. Nó biểu diễn một ứng dụng.

Phần tử **package** là phần tử con của struts. Nó biểu diễn một Module của ứng dụng. Nói chung, nó kế thừa struts-default package, là nơi định nghĩa nhiều Interceptor và Result type.

Phần tử **action** là phần tử con của package. Nó biểu diễn một action để được triệu hồi cho request vừa đến. Nó có các thuộc tính tên, lớp, và phương thức. Nếu bạn không xác định thuộc tính name, theo mặc định thì phương thức execute() sẽ được triệu hồi cho lớp Action đã cho.

Phần tử **result** là phần tử con của action. Nó biểu diễn một view (đây là kết quả) mà sẽ được triệu hồi. Nó có các thuộc tính name và type, cả hai là tùy ý. Nếu bạn không xác định name, thì theo mặc định success được xem như là tên mặc định của nó. Nếu bạn không xác định thuộc tính type,

thì theo mặc định dispatcher được xem như là Result Type mặc định. Chúng ta sẽ tìm hiểu về result type trong các chương sau.

struts.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
<package name="default" extends="struts-default">

<action name="product" class="com.vietjack.Product">
<result name="success">welcome.jsp</result>
</action>

</package>
</struts>
```

Bước 6: Tạo các thành phần view

Đây là thành phần view hiển thị thông tin của action. Ở đây, chúng ta sử dụng struts-tags để lấy thông tin. Thẻ s:property trả về kết quả cho name đã cho, được lưu trữ trong đối tượng Action.

welcome.jsp

```
<%@ taglib uri="/struts-tags" prefix="s" %>

Product Id:<s:property value="id"/><br/>
Product Name:<s:property value="name"/><br/>
Product Price:<s:property value="price"/><br/>
```

Bước 7: Tải jar file

Để chạy ứng dụng này, bạn cần có struts 2 jar file. Tại đây, chúng ta cung cấp mọi jar file cần thiết cho Struts 2. Bạn [tải tại đây](#) và đặt các jar file này trong lib folder của project.

Bước 8: Khởi động Server và triển khai project

Để tạo ứng dụng Struts, bạn có thể sử dụng các IDE như eclipse, myeclipse, netbeans, RAD, JDeveloper, ... Chương tiếp theo sẽ trình bày cách tạo chương trình Struts 2 đầu tiên bởi sử dụng myeclipse IDE.