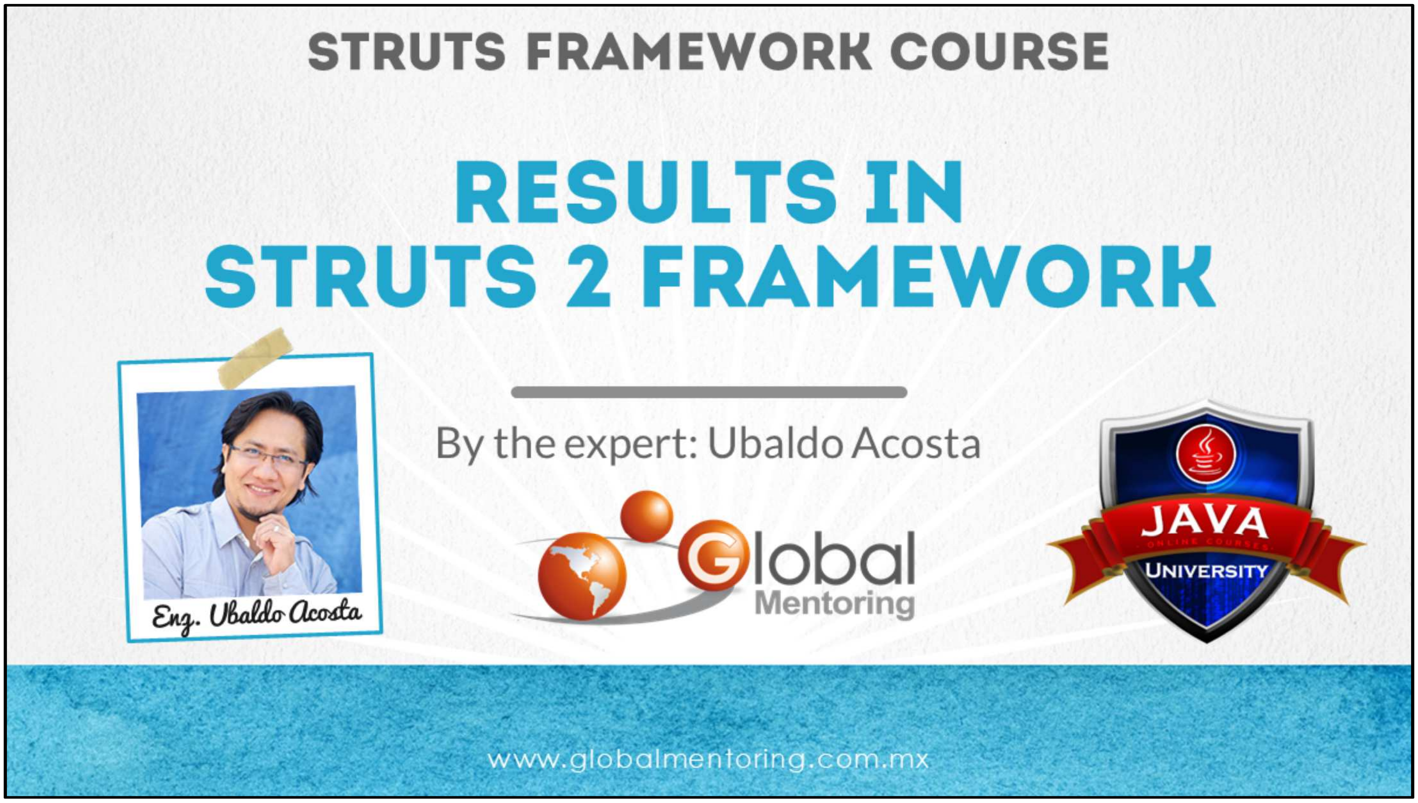


STRUTS FRAMEWORK COURSE

RESULTS IN STRUTS 2 FRAMEWORK

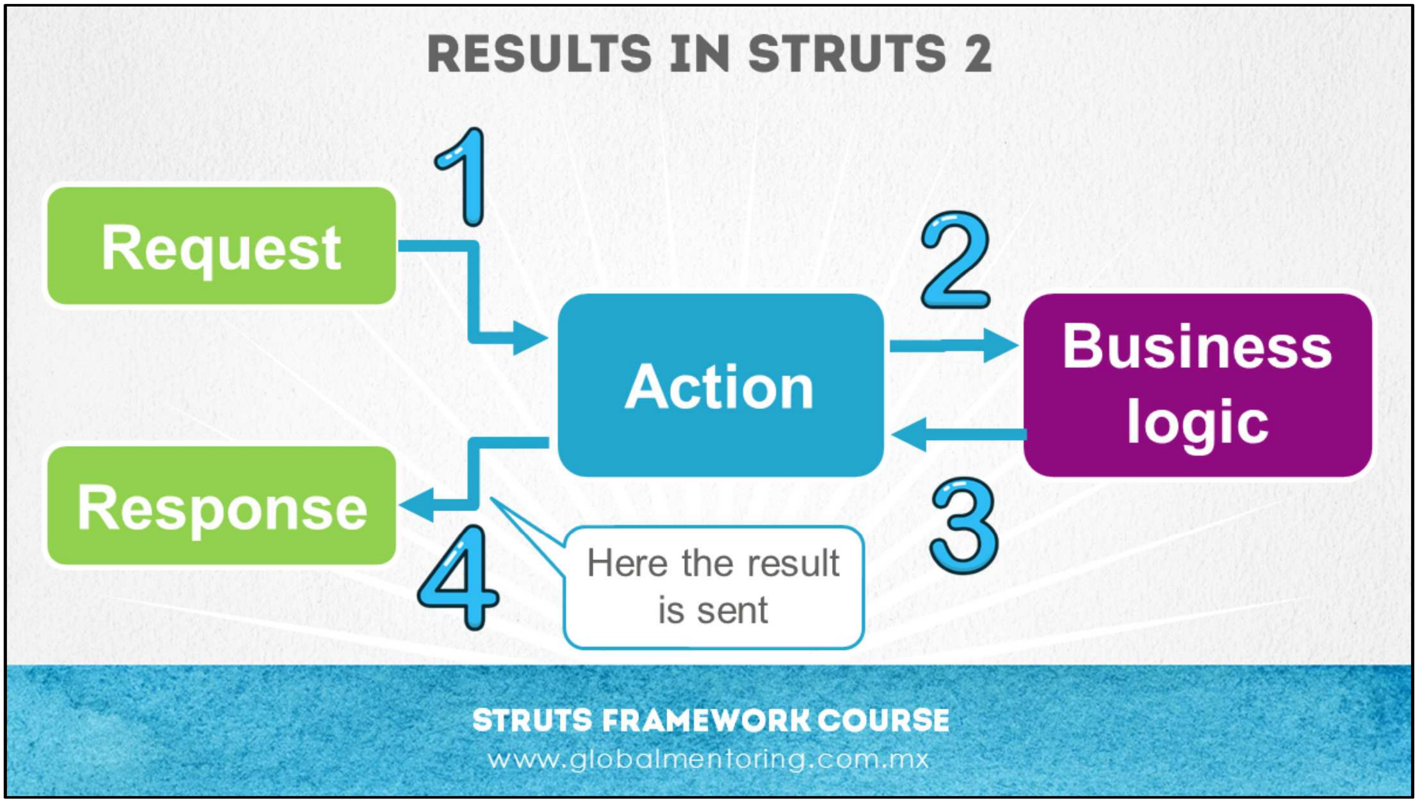
By the expert: Ubaldo Acosta

www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you again.

In this lesson we are going to study the topic of Results in Struts 2.

Are you ready? Come on!

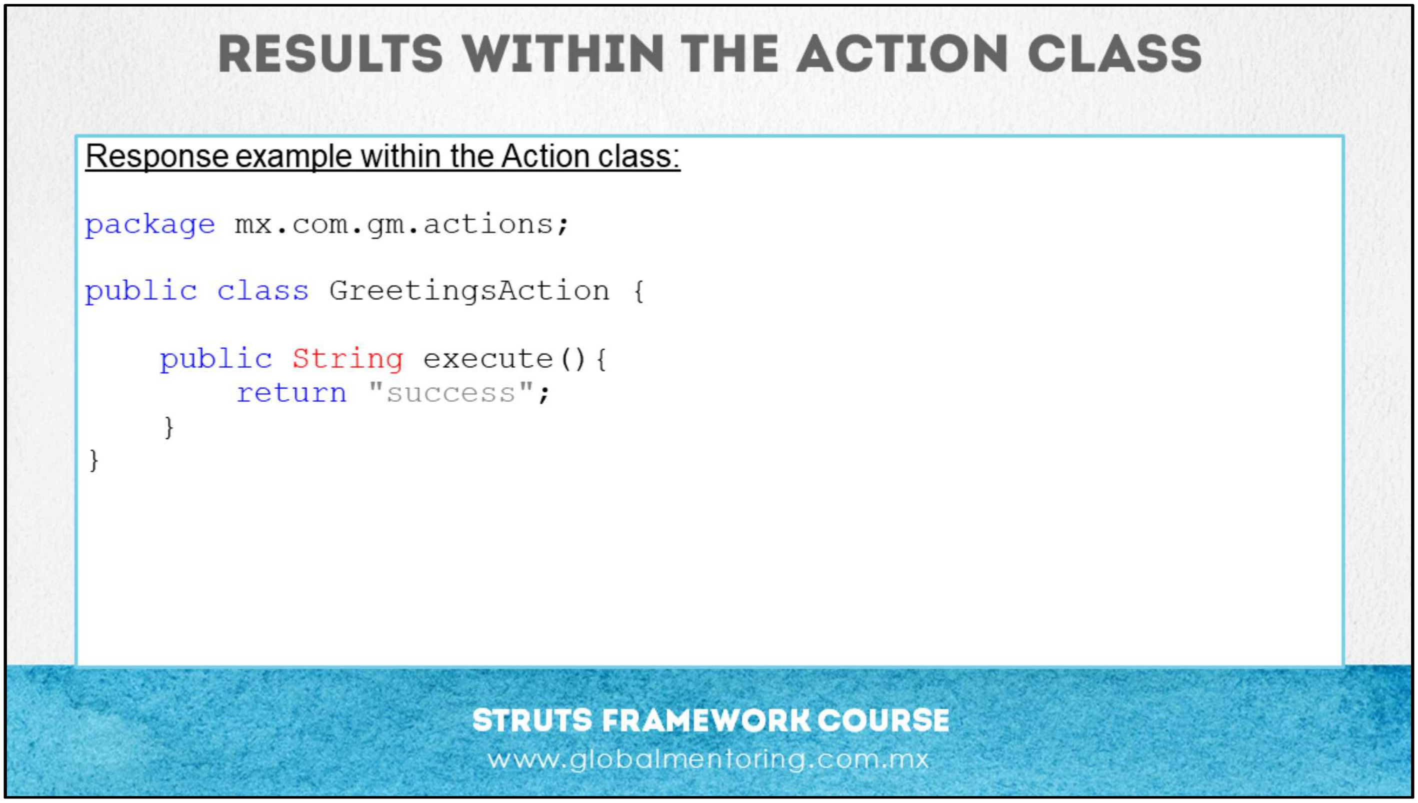


In this lesson we will see the topic of managing results in Struts 2.

We have already seen the general stages through which a petition passes in Struts. Now we have to go deep in step 4, where the result of executing a certain action in Struts 2 will be specified.

A result is defined from a string or String and depending on this string, the type of result that will be sent to the user is selected. Commonly the answer is a JSP, however there are other types of results, such as a Servlet, a document or a Tile (the Tiles is a template used to define the parts of a view, such as header, body, footer, etc and thus be able to make the sections of said template reusable).

However, the most common is that from the chain or String that we return, we select the JSP to deploy to the client. This configuration can be added to the struts.xml file within the configuration of our action by means of the <result> tag or we can use the concept of annotations within the respective Action class. The concept of annotations we will see later, but the annotation to use is @Result or if we want to define several results we will use the annotation @Results and within this several annotations of type @Result separating each one by keys {}.



RESULTS WITHIN THE ACTION CLASS

Response example within the Action class:

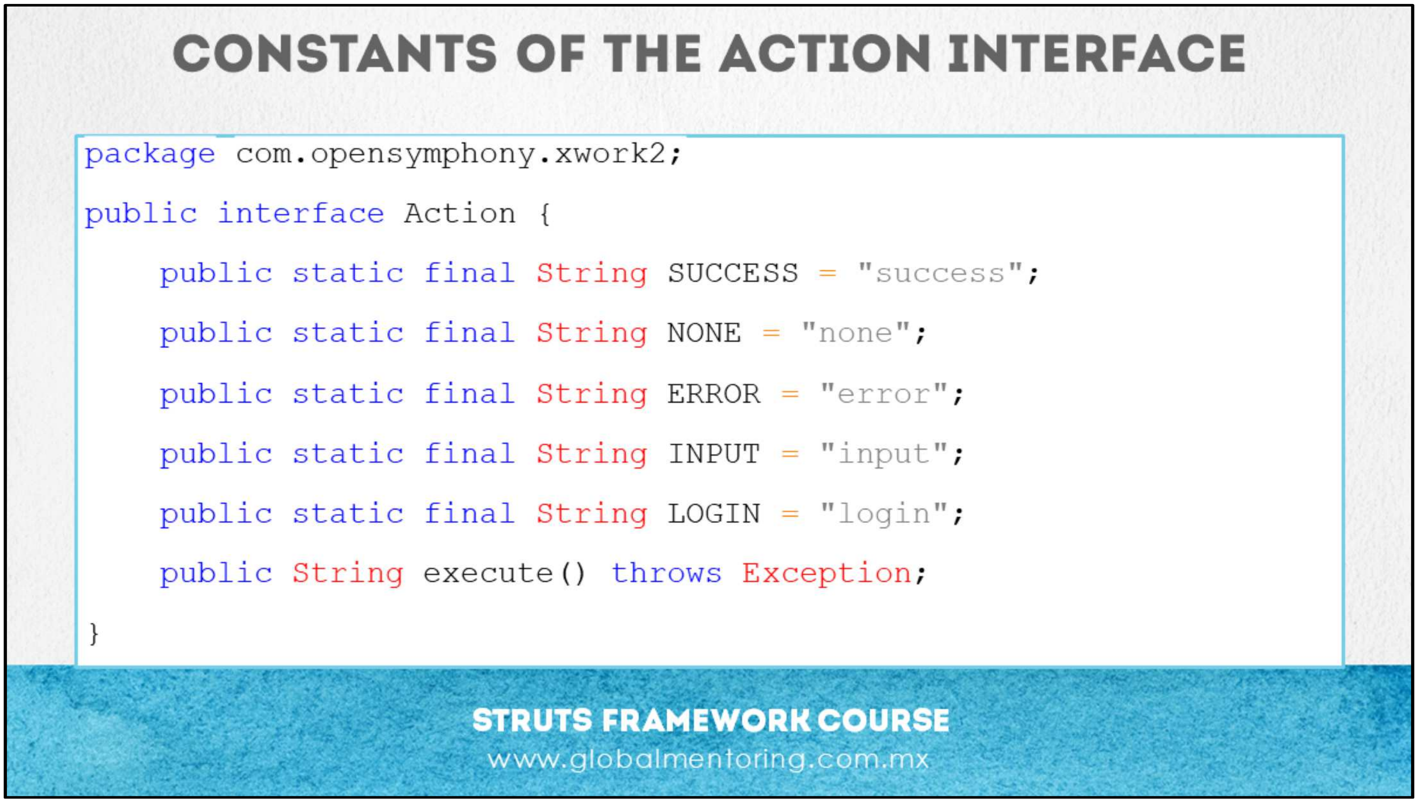
```
package mx.com.gm.actions;

public class GreetingsAction {

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

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

We can see in the code that the answer within the execute method is the string of "success" in this case. And with that chain is that we will select the view to show the user. This response, as we have said, can be configured in the struts.xml file or through the @Result annotation within the Action class, either at the class level or at the execute () method level.

We will see later how we must configure the struts.xml file to handle the response.



CONSTANTS OF THE ACTION INTERFACE

```java
package com.opensymphony.xwork2;
public interface Action {
    public static final String SUCCESS = "success";
    public static final String NONE = "none";
    public static final String ERROR = "error";
    public static final String INPUT = "input";
    public static final String LOGIN = "login";
    public String execute() throws Exception;
}
```

STRUTS FRAMEWORK COURSE
www.globalmentoring.com.mx

These are the constants that we can use by default when extending the Action interface. However, we note that in the end they are simply a string, so we can use the constant described in this interface or directly put the string we want to return so that the response to be displayed to the client is selected (the JSP or view to be displayed).

The string of success is the answer by default, and if nothing else is indicated in the configuration of the response of the action in the struts.xml file then the response indicated in the <result> tag will be displayed or it will be selected the view that matches the name of the action according to the conventions that we have previously commented on Struts 2.

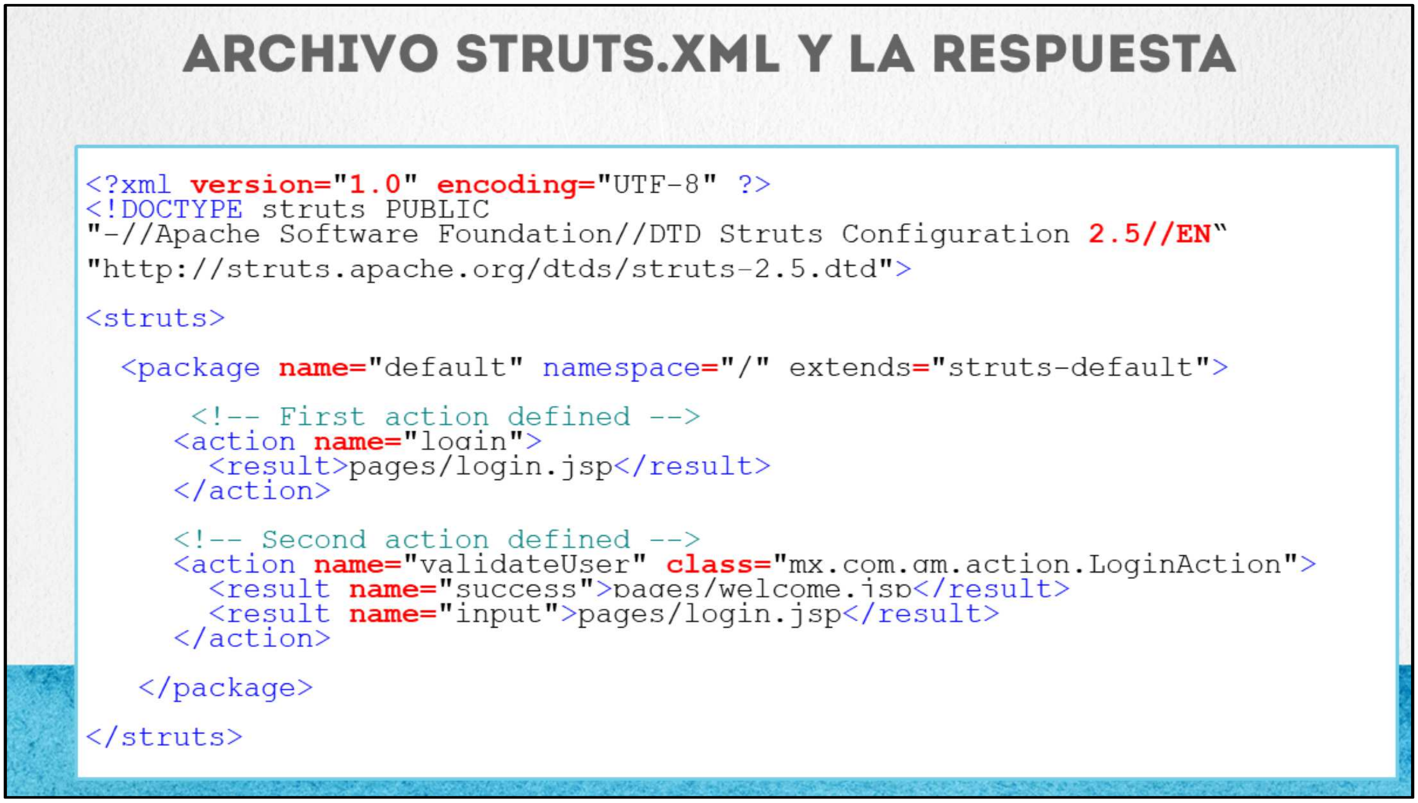
The input string is normally used to return to the view where some data of some form was captured, and we indicate that we return to that form whatever it may be. It is only a convention, so we can use this chain or any other to return to a form that we need to show the user again, such as capturing a user's data.

The error string is normally used to indicate the view to be displayed in case of error.

The login string is normally used to indicate that we are going to show the view of the page that works as a system login, where we request the user and password values.

The string of none is used to indicate that no action is executed, it is sometimes used in AJAX type requests where it is not necessary to go to another view, but to continue using it.

Let's see below how to configure our struts.xml file to handle the response.



ARCHIVO STRUTS.XML Y LA RESPUESTA

```
<?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>

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

       <!-- First action defined -->
      <action name="login">
        <result>pages/login.jsp</result>
      </action>

      <!-- Second action defined -->
      <action name="validateUser" class="mx.com.gm.action.LoginAction">
        <result name="success">pages/welcome.jsp</result>
        <result name="input">pages/login.jsp</result>
      </action>

   </package>

</struts>
```

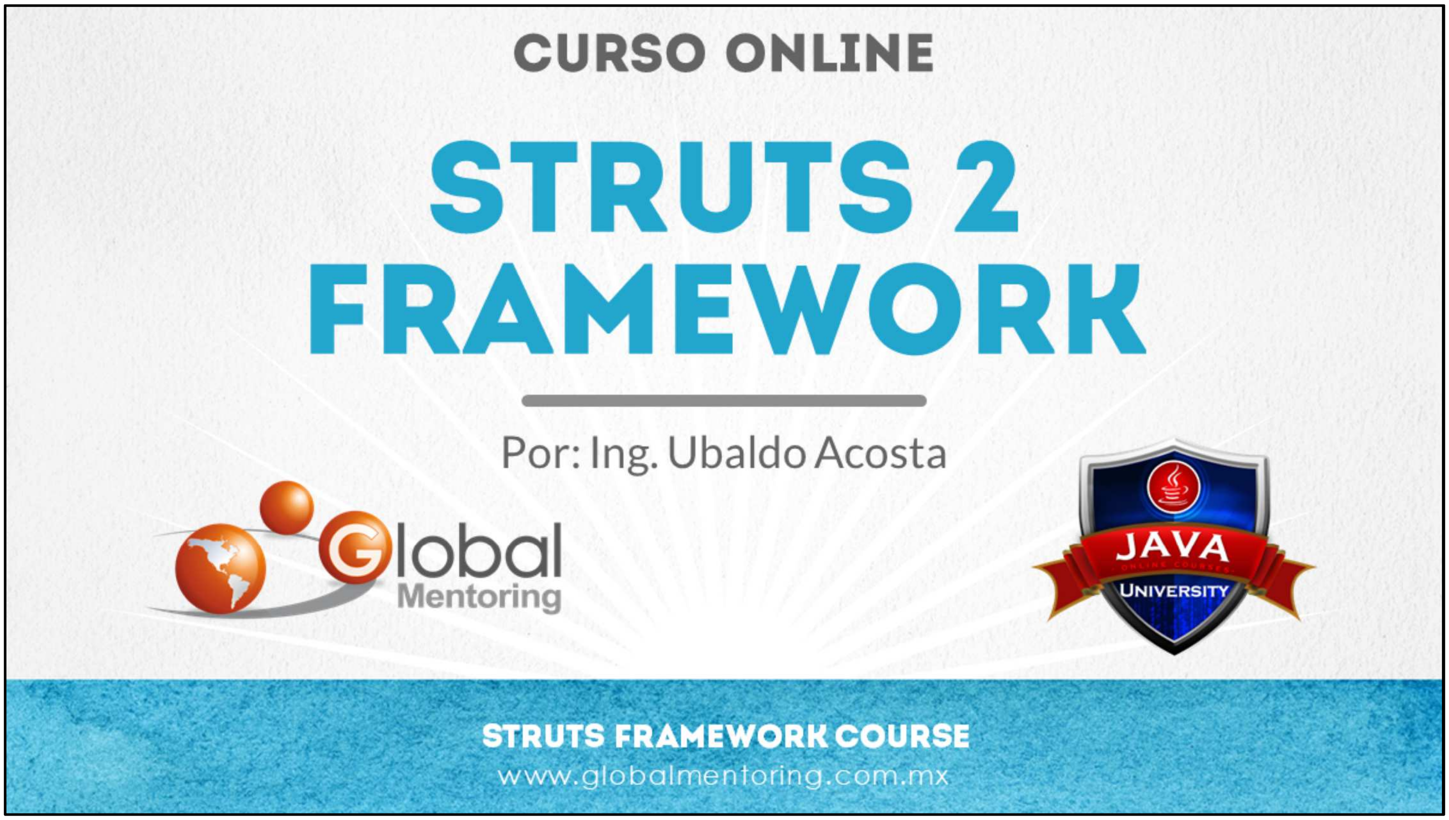
We can observe the configuration of our action and how we should add the possible answers that our action can throw, because it is a chain, we can return any chain, but each of these chains must be mapped in the action within the struts.xml file to that Struts knows which is the view that must show as a response to the executed action.

We can observe in the first action for the path="/user", it is NOT necessary to execute a class of type Action to obtain a result and forward to a new JSP. In this way we will continue to respect the MVC design pattern when using the steps of the Struts framework to make a call to a JSP without having to go through an Action, but we will go through the Struts framework and the listeners that are configured in the option of extends = "struts-default". In this way we will be executing the Struts framework even in simple cases like calling directly to another JSP, in this way we do not break with the MVC pattern. On the other hand, if we do not have more options in our action, it is considered that the only <result> that has been configured is the one that will be executed. Therefore regardless of the chain you return, the login view will be login.jsp

In the second action we observe that for the path "/validateUser" it executes the class LoginAction. This class in its execute method can return two values when doing the return of the string. The first chain that is validated is if it returns "success", in which case the view of welcome.jsp will be used. In the second case, if the input string is returned as return of the execute method, the login.jsp view will be used. In this way we can observe that in case of returning several values, we can indicate different results and therefore, select different views depending on the result sent by the execute method.

We can add this same configuration to the Action class using annotations, however it is necessary to know both configurations in case we find one code or another.

Here is an example of response management using Struts 2.



CURSO ONLINE

STRUTS 2 FRAMEWORK

Por: Ing. Ubaldo Acosta

Global
Mentoring
JAVA
UNIVERSITY

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx