

STRUTS FRAMEWORK COURSE

ACTIONS WITH STRUTS 2 FRAMEWORK



By the expert: Eng. Ubaldo Acosta



www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you again.

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

Are you ready? Come on!

ACTIONS WITH STRUTS 2 FRAMEWORK

- An action is one of the most important parts of Struts
- An action represents the Controller in the MVC pattern
- There are different types of Action in Struts, depending on what you want to do
- It is not necessary to extend any interface, but it is possible to choose among several that we will see next.

STRUTS FRAMEWORK COURSE

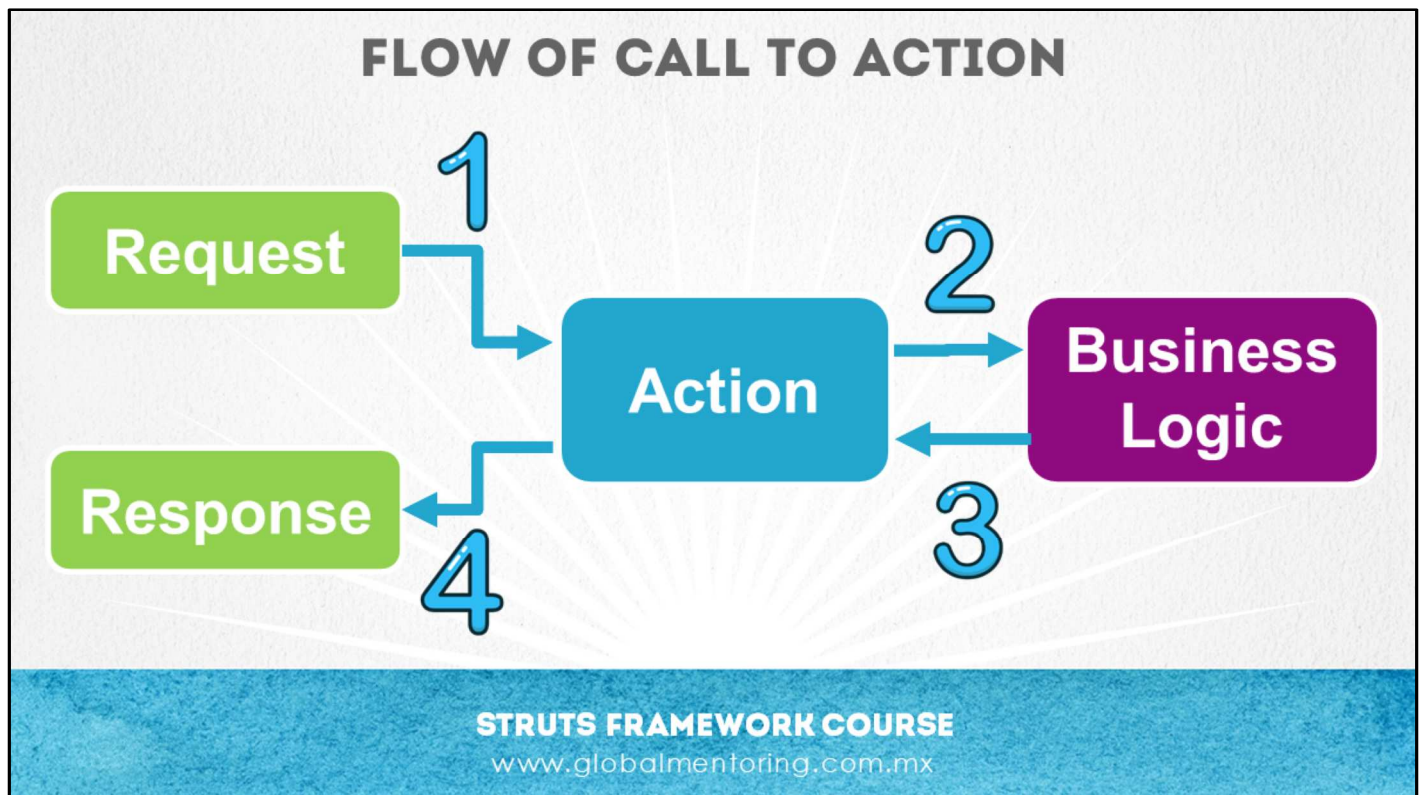
www.globalmentoring.com.mx

In this lesson we are going to study the topic of Actions in Struts 2. Actions are a very important topic in Struts 2, because everything that happens in the framework goes through the Action type class that we create. So we must have a very good understanding of what actions are and how we can use them to successfully build an application with Struts 2.

An action answers the question of What are you trying to do ?, making reference to the action that you want to execute in the Web application. From the point of view of the MVC (Model-View-Controller) pattern, the action plays the role of Controller, so it is the part that orchestrates many of the functions that the framework performs.

There are different types of actions, which we will see later.

When we create an Action class, it is not necessary to extend it from any other class or interface, as was necessary in Struts 1. However, in some cases it is recommended depending on the type of action that we need to implement. Next we will see the types of Shares that we have in Struts 2.



Let's now see the flow of call to an Action in Struts 2. We can see that the action is the central part of the MVC design pattern, where it plays the role of Controller, since the Action passes the request and the response to the user. The action is also responsible for tasks such as processing user data, validating them, applying the concept of internationalization if necessary, communicating with the business logic layer, among several other tasks.

So understanding and correctly implementing the issue of actions in Struts is crucial. However, the good news is that creating an action in Struts is extremely easy, and in many cases extending from the `ActionSupport` class (which we will see later) will be sufficient for most of the cases.

We observe in the figure that everything starts with the request of the user in the Web browser (request). This request is processed by the Action class that has been configured to respond to the path contained in the URL that was sent by the Web browser.

Once the Action class receives the request, it is common for several tasks to be performed, such as:

- Reception of data (HTML form parameters)
- Data validation
- Internationalization Management (convert strings or texts to other languages)
- Initialize the model (either attributes of the Action class or independent Domain classes)
- Call business logic methods
- Determine the next view to show (result)
- Among several other tasks.

Once the action has finished processing the data, and perform several of the tasks mentioned above, then a result (result) returns, which selects the view to be displayed again to the user, and this is the response process (response)

CREATING AN ACTION TYPE CLASS

Example of an Action type class in Struts 2 without implementing interfaces:

```
package mx.com.gm.actions;  
  
public class GreetingsAction {  
  
    public String execute() {  
        return "success";  
    }  
}
```

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

Let's see now how to create a class of type Action.

As we saw in our first exercise, we can implement an action in Struts 2, simply adding the public method called `execute`, which does not receive any parameters, and returns a `String`, which will indicate the result of the next view or page to be displayed.

This is a very simple way to add an action in Struts. However it is not the most recommended, later we will see another option that would be the one we will most frequently use.

Remember the conventions that we must apply if we want to avoid the configuration in the `struts.xml` file

The action must be found in a Java package, which includes any of the following names: `action`, `action2`, `struts` or `struts2`. The class must also end with the word `Action`, and this will be enough to not have to configure our action in the `struts.xml` file, but only by managing the conventions can we use our action.

CREATING AN ACTION TYPE CLASS

Example of an Action type class in Struts 2 implementing interface:

```
package mx.com.gm.actions;  
  
import com.opensymphony.xwork2.Action;  
  
public class GreetingsAction implements Action {  
  
    @Override  
    public String execute() {  
        return "success";  
    }  
  
}
```

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

We can see that we can also implement a Struts 2 action by implementing the Action interface, which belongs to the com.opensymphony.xwork2 package

In this way we should only overwrite the execute method as shown.

However this is not the most recommended way, as we will see other ways to create an action in Struts 2 later.

CONSTANTS OF THE ACTION INTERFACE

```
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;
}
```

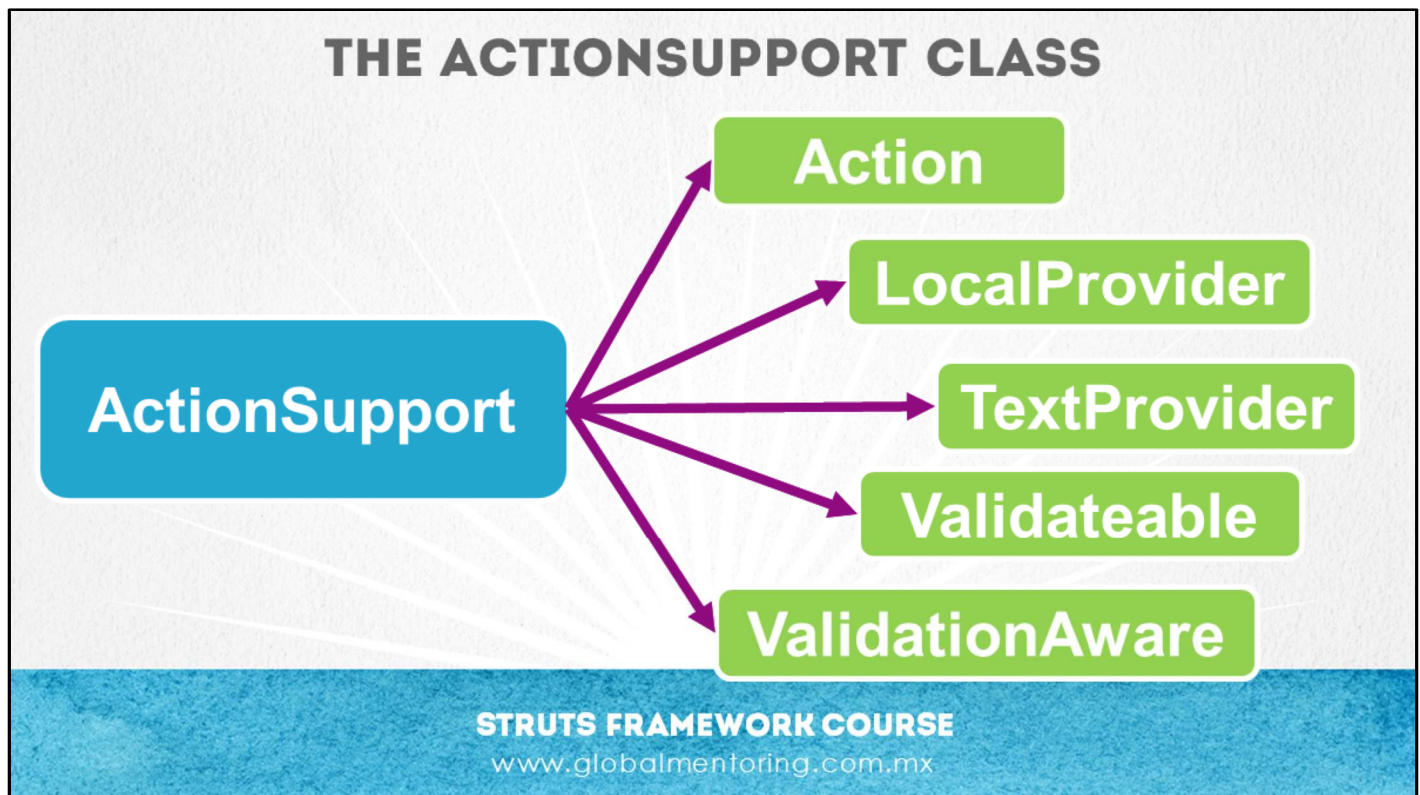
STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

We can see that as part of using the Action interface, in addition to implementing the execute method, we also have several String type constants that we can use to indicate the result of our execute method.

The constant that we will most use is that of SUCCESS, which automatically indicates that the result of the action is the one associated with the action either in the struts.xml file or the view corresponding to the name of the JSP associated with the action according to the concept of conventions that we studied previously.

However, we can use any of the constants shown to indicate the result or view we want to use.



In the figure we can see the ActionSupport class, and all the interfaces that this class implements. These interfaces are the types that we have available in Struts 2 to create an Action type class.

So because the ActionSupport class implements all these interfaces, this is the class we recommend to extend and use when we create our actions in Struts 2, since it solves most of the requirements.

Some of the requirements that this class covers are:

- Data validation
- Location (Messages and Internationalization)
- Management of Constants to indicate the result (next view to show)
- Among several other features

So this class is what we will extend in most cases when we create actions in Struts 2.

CHARACTERISTICS OF THE CLASS ACTION SUPPORT

```
public class ActionSupport
    implements Action, Validateable, ValidationAware,
        TextProvider, LocaleProvider, Serializable {

    // We can overwrite methods like validate for data validation:
    validate() {
        ...
    }

    // Or methods like getText for the concept of internationalization:
    getText() {
        ...
    }
}
```

STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx

As we have said, the ActionSupport class is the class that we will most commonly use to create our actions in Struts 2. As we can see, it implements the interfaces of all the cases that we could find in Struts 2.

Implements interfaces such as: Action, Validateable, TextProvider and Localeprovider.

And basically these interfaces allow us to handle in a very simple way concepts such as data validation of the HTML form, handling of labels or messages using the concept of internationalization, and we can even access the constants that we have already commented on in the Action interface.

In such a way that when we create our actions it is advisable to extend the ActionSupport class.

Next we are going to create an exercise to put this concept into practice.

ONLINE COURSE

STRUTS 2 FRAMEWORK

By: Eng. Ubaldo Acosta



STRUTS FRAMEWORK COURSE

www.globalmentoring.com.mx