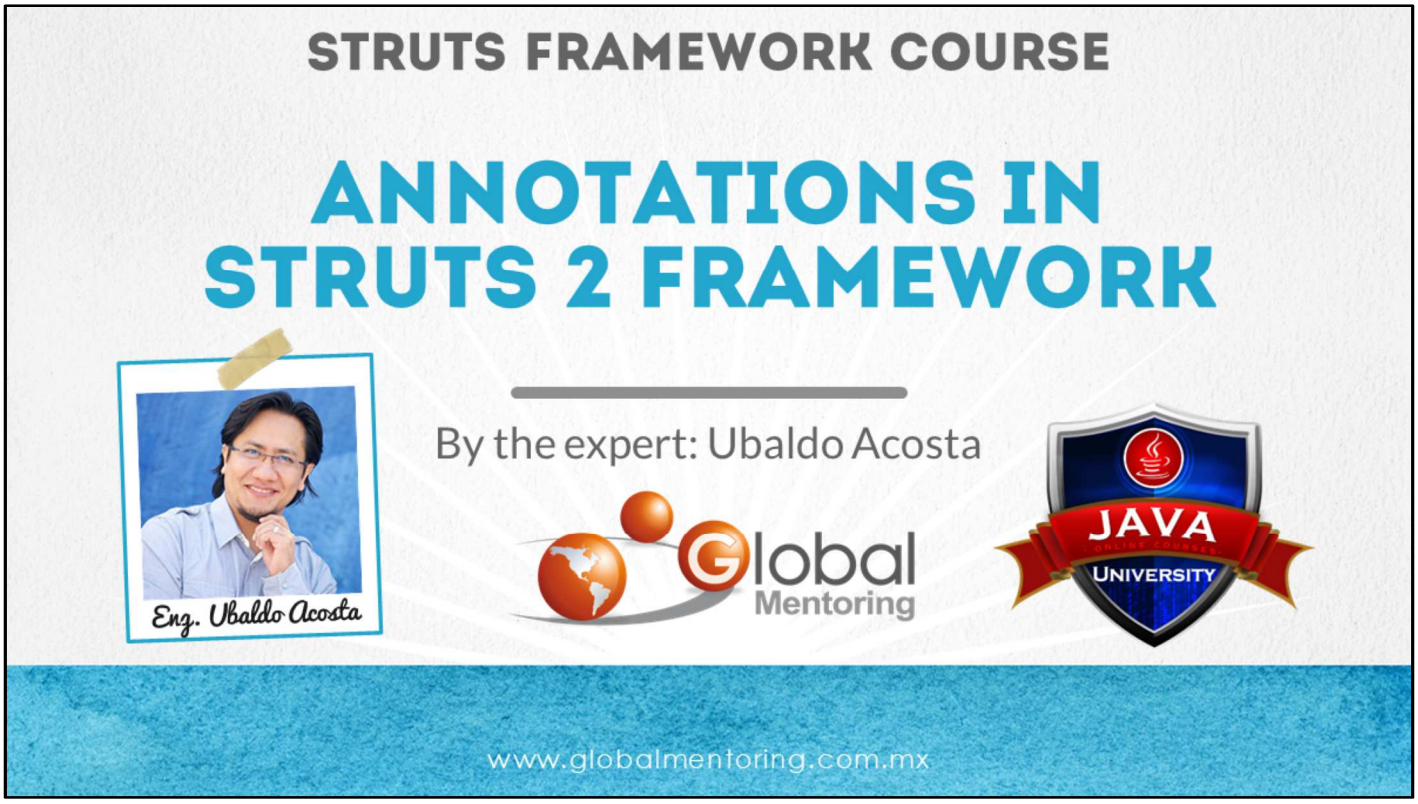
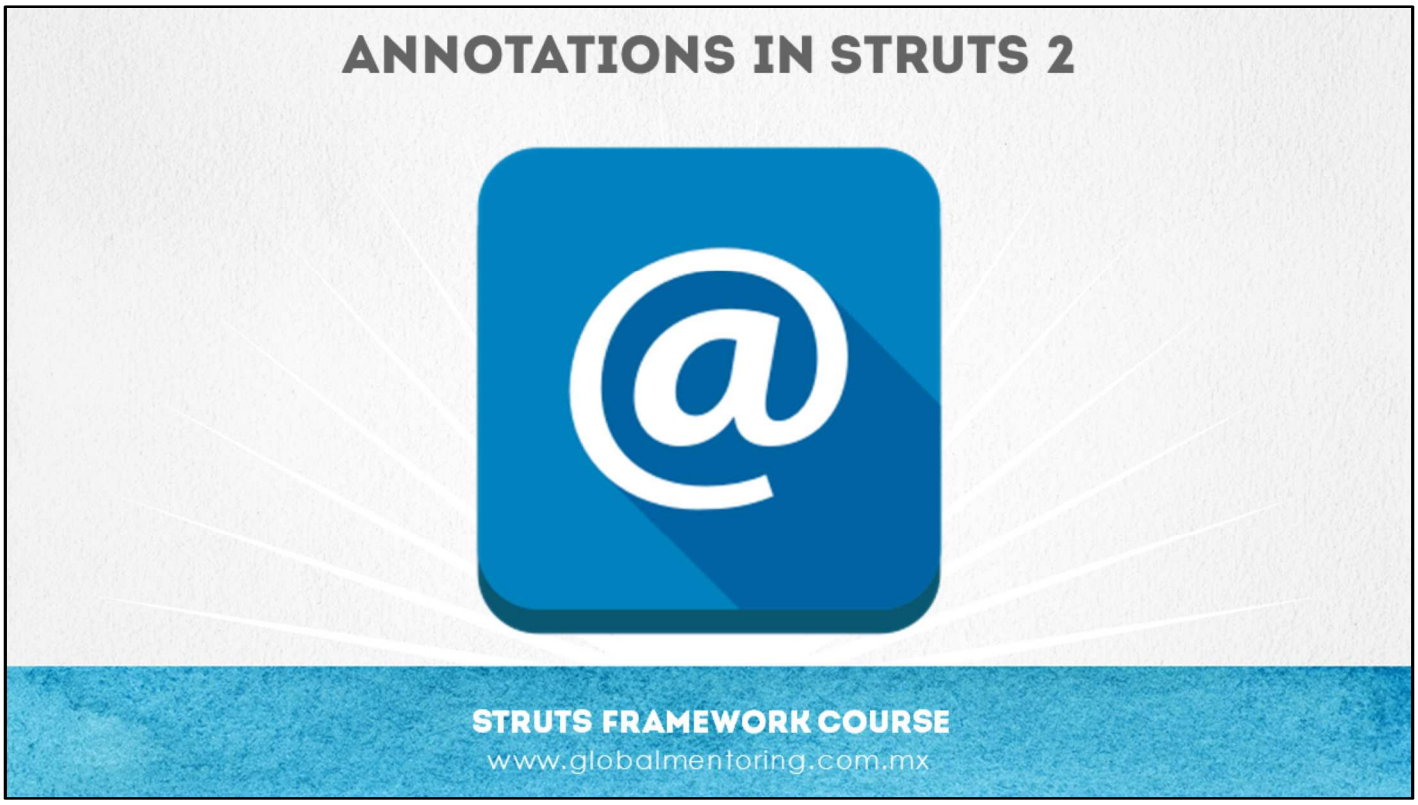


Hello, Ubaldo Acosta greets you again.

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

Are you ready? Come on!



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

As we have seen, to replace the configuration we have used in the struts.xml file we have used the theme of conventions in Struts 2. Otherwise we must declare our action, as well as the possible results in the struts.xml file.

However there is another way to make this configuration and it is directly in our classes of type Action using the annotations of Struts 2

The annotations in the Action class can be used to replace the declaration of actions in the struts.xml file, but also for practically many of the elements that we use in our actions, such as the handling of results, types of results, validations, etc.

With the use of annotations we can practically avoid the use of the struts.xml file, so it is up to the architect or programmer if the configuration of the actions in the struts.xml file or within the same Action type classes of Struts is used. 2.

Let's see some examples for the use of annotations, instead of the configuration that we have applied in the struts.xml file.

ANNOTATIONS OF TYPE @ACTION

Example of using annotations of type @Action within the Action class:

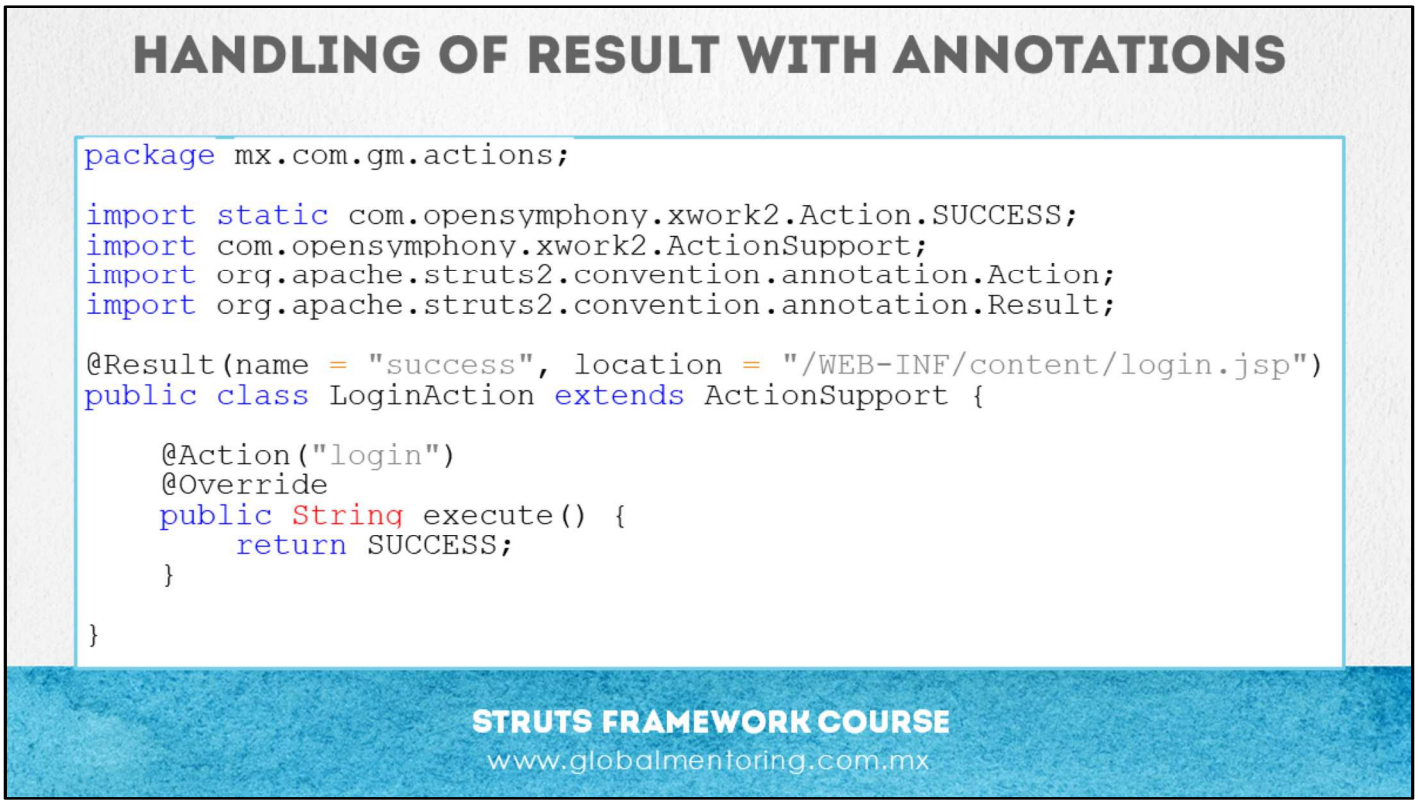
```
package mx.com.gm.actions;

import org.apache.struts2.convention.annotation.Action;

public class GreetingsAction {

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

STRUTS FRAMEWORK COURSE
www.globalmentoring.com.mx

We can observe in the code we are using the annotation @Action(“GreetingsAction"), this means that the path to which this class of type Action will react will be equal to: /GreetingsAction

This way it is no longer necessary to add the configuration to the struts.xml file.

Let's see some examples of the handling of results in Struts 2 with annotations.



In a similar way, we can add the use of annotations of type @Result to our class of type Action. If we have multiple values we must use the @Results annotation, and then add each of the annotations of type @Result that we want to add separately by comma.

This annotation is added at the level of the class or at the level of the method. But if it is added at the level of the method it must be within the @Action annotation that was previously defined in the method.

We see in the code an example of the use of this annotation at the class level. And what we are specifying is that once our execute method marked by the @Action annotation returns the string of success, then review what is the result that has been configured by means of the annotation @Result. And through this annotation is that we can specify which is the view that will be displayed associated with the result thrown.

Let's see below an example of several possible outcomes.

VARIOUS TYPES OF RESPONSES IN THE ACTION CLASS

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

import com.opensymphony.xwork2.ActionSupport;
import org.apache.struts2.convention.annotation.Action;
import org.apache.struts2.convention.annotation.Result;
import org.apache.struts2.convention.annotation.Results;

@Results({
    @Result(name="success", location="/WEB-INF/content/welcome.jsp"),
    @Result(name="input", location="login", type="redirectAction")
})
public class ValidateUserAction extends ActionSupport {

    //Omit the declaration of the user variable


    @Action("validateUser")
    @Override
    public String execute() {
        // If you are a valid user, we show the welcome.jsp page
        if ("admin".equals(this.user)) {
            return SUCCESS;
        } else {
            //If you are an invalid user, we return to login
            return INPUT;
        }
    }
}
```

As we have said we can also have several types of responses, we can see an example in the code.

We observe in the code shown that we are validating whether the user variable contains the string "admin", in which case we return the string of "success", otherwise we return the string of "input". So our execute method can return different values according to the validation that is done in the method.

Once we have returned some of these values, we review the @Results annotation and to this annotation we add the different cases that our execute method can return. For each of the cases we will separate them by a comma and use the annotation @Result as we have previously used it.

However we see in the case of input, that we can not only indicate a JSP as the view to use, but we can again indicate a path that leads us to execute an action again, and to change the URL of the new location or action executed we use the type = "redirectAction" attribute, with this what we are doing is a redirection to the indicated path and that also changes the URL that will be displayed in the browser, otherwise the path of "validateUser" will be shown, since that is the path that was executed to enter this action as indicated in the @Action annotation in the execute method.

In this way we can see how we have replaced the configuration of our struts.xml file and now we have used annotations for the configuration of our Action class and the handling of results.

There are more annotations, but these are the most used. We are going to make an example below to implement the use of annotations with Struts 2 within our Action class.