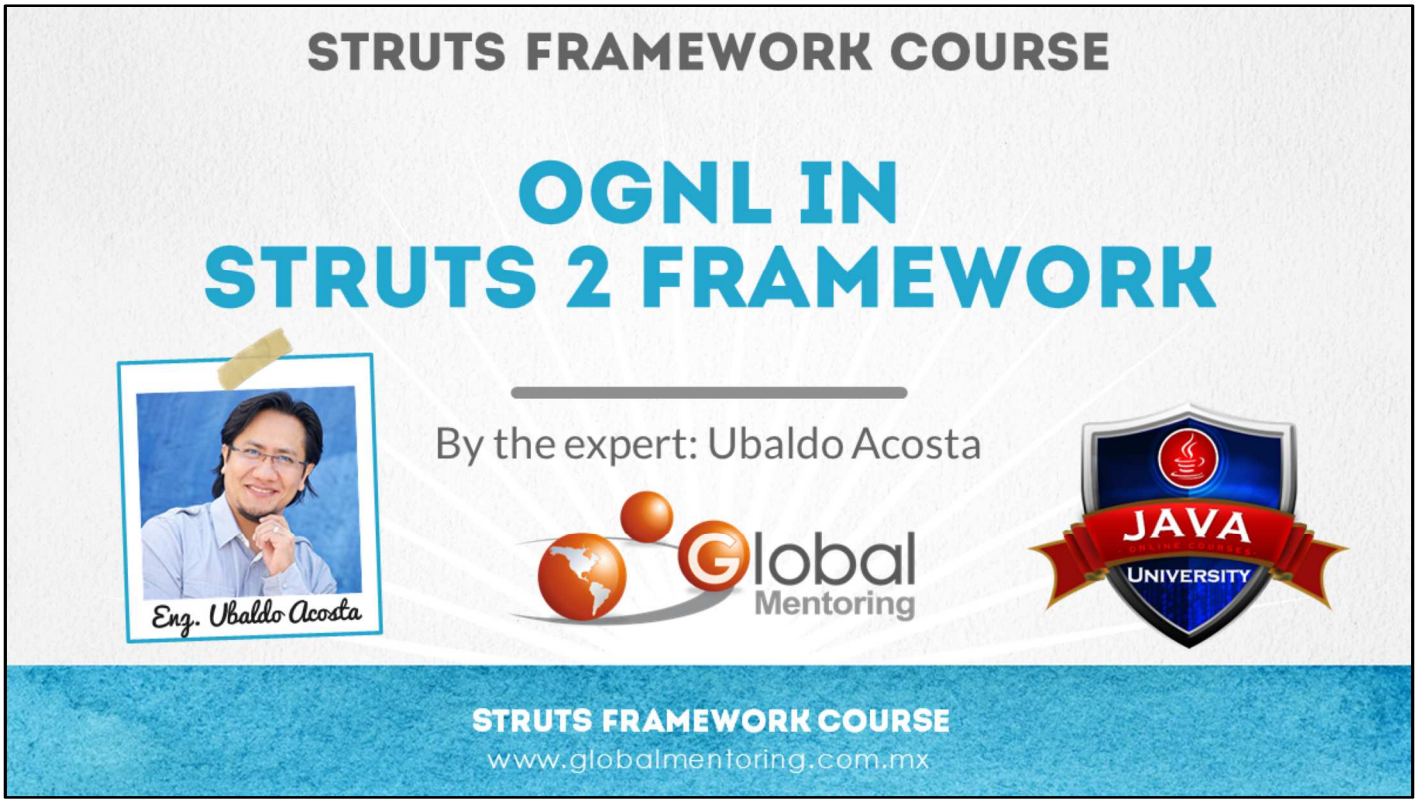
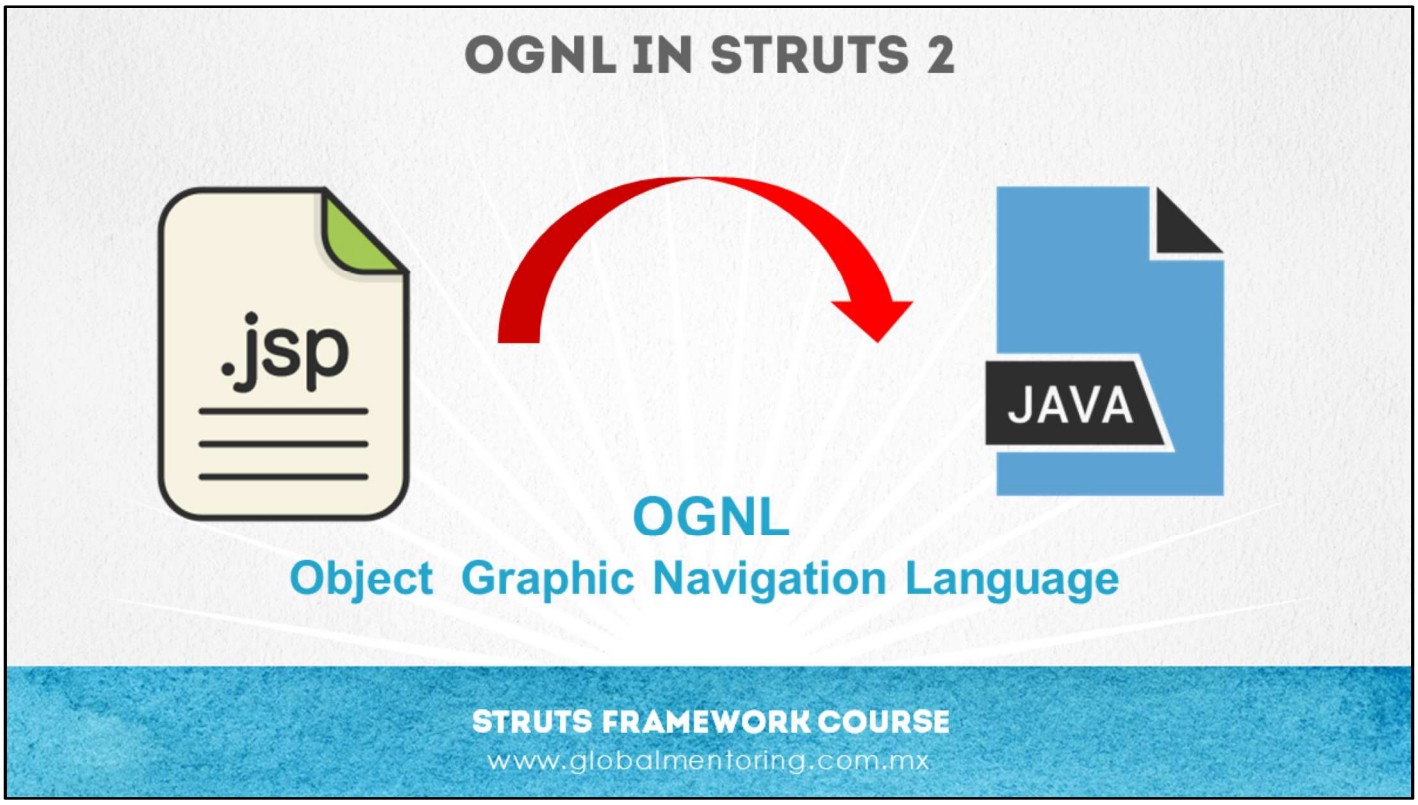


Hello, Ubaldo Acosta greets you again.

In this lesson we are going to study the topic of Object Graphic Navigation Languaje (OGNL) in Struts 2.

Are you ready? Come on!



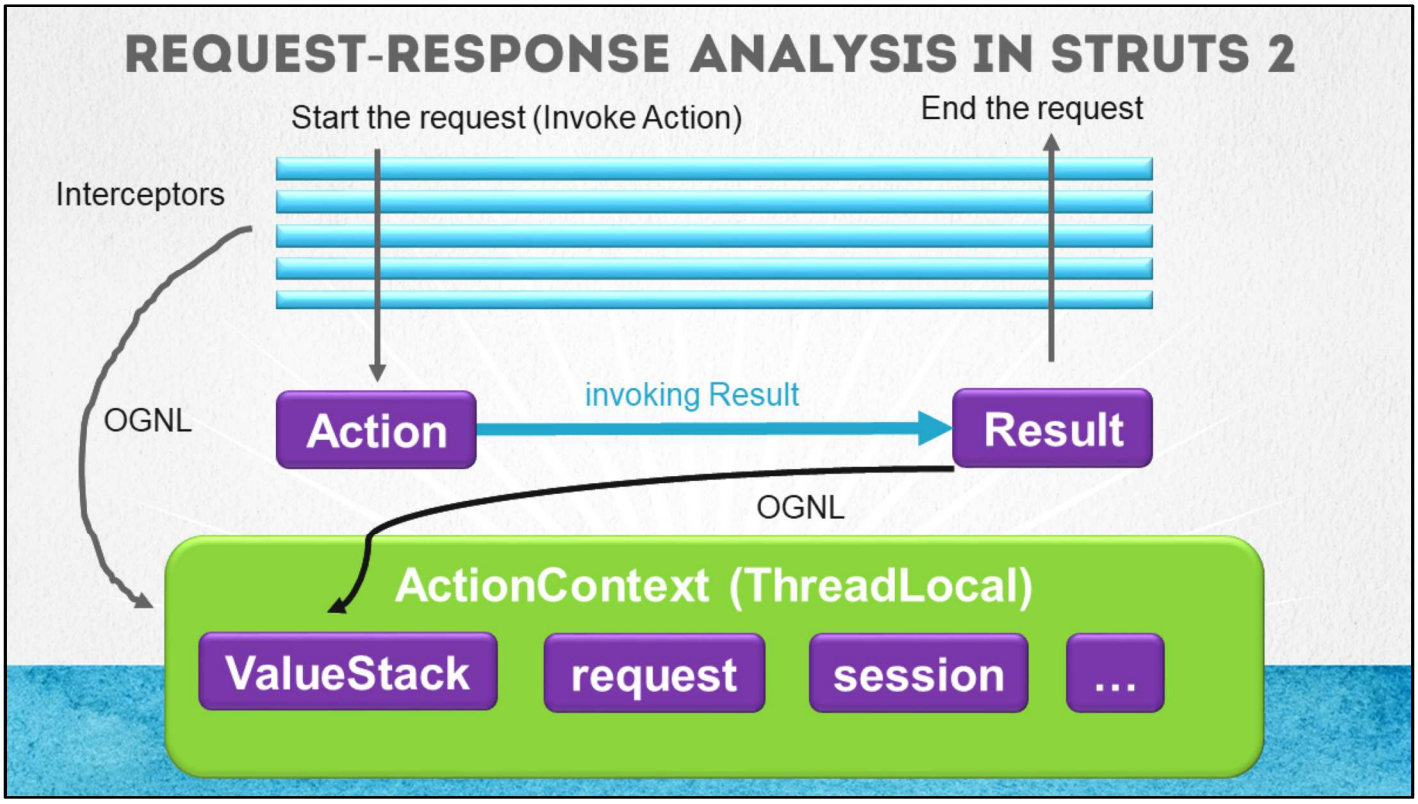
In this lesson we will see the topic of managing OGNL in Struts 2.

This issue we have already worked with OGNL, however in this lesson we will formalize this concept so useful in Struts 2.

ONGL stands for Object Graphic navigation Language, and is a tool that emerged in Struts 2 with the aim of being able to access our Java objects from the view (JSP's).

By this we mean that the attributes of our Action classes or any other class that is added to the Struts variables stack, we can access it directly from our JSP's in several ways, either from Struts tags, using Expression Language, Scriplets, or any another variant of our view to be able to access the properties of our Java classes.

One of the great advantages is that there is no need for us to perform an instance of the Java classes manually, but Struts is smart enough to detect if the variable we are wanting to use already exists in the StackValue of Struts variables or do not. And in case the variable does not exist, then it creates it and assigns it the values of our view as appropriate.



The Struts 2 framework is more than just the implementation of the MVC design pattern. Struts 2 is built on a very robust architecture as we can see in the following diagram.

In the figure we can see the flow of a request in Struts 2 in more detail. On the one hand we observe that when the request starts, this request passes through several Interceptors. By interceptor we refer to the concept of Interceptors of the Servlets API, which basically can intercept and modify a request and its response, in the same order in which the request is processed, the response is processed in the inverse direction in how it is processed the request initially.

These interceptors can access the ActionContext, which is an object of the Struts API that contains all the variables that are used in the application, including ValueStack, request, session, application, etc.

To access these values is to use the concept of OGNL, indicating which variable we want to access, either from the interceptors, or for example also from the JSP view or any other technology that is used to create the view when process the result of the request.

The ValueStack section is where all the data associated with a particular request is stored. OGNL is a tool that allows us to access the data that the Struts framework has added to the ValueStack.

Once this concept is clarified, we can go on to an example of how to use the ValueStack data, we will see that it is very simple, and that basically the framework has greatly simplified the process of accessing the data we use from our view.

OGNL IN STRUTS 2

Example of use of OGNL. Definition of variables within the Action class :

```
public class PersonAction {

    private String firstName;

    private String lastName;

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

Example of using OGNL within the JSP view associated with the action shown:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib  prefix="s" uri="/struts-tags" %>
<!DOCTYPE html>
<html>
    <head>...</head>
    <body>
        <h1>Person Form with OGNL</h1>
        <s:form>
            <s:textfield label="FirstName" name="firstName" />
            <s:textfield label="LastName" name="lastName" />
            <s:submit value="Send"/>
        </s:form>
...
```

We can see in the code that we are using an Action class called PersonasAction.java, and on the other hand we are using the respective JSP view called personas.jsp

In order to access the attributes of the class, we will support the ValueStack where we have commented that all the variables of the current request are stored.

In the JSP we can see that to access the attributes it uses OGNL instead of JSTL of JSPs, in this way the framework facilitates access to the variables.

We can see this in the <s: texfiled> tag, however OGNL is much more powerful than accessing simple properties, you can also create and access more complex attributes, for example:

<s: textfield label = "No Street (Address)" name = "person.address.streetNumber" />

In this example the textfield tag is accessing the property streetNumber, which is an attribute of the object Address, which in turn is an attribute of the Person object, so we can see that OGNL is smart enough to access this type of properties simply using the described syntax. All these values and objects are stored in the Struts ValueStack so that they can be accessed automatically by each request.

Let's see below an example to implement the concept of OGNL.