



## Struts 2 OGNL



- Una de las tareas más importantes en una aplicación web es la transferencia de datos
  - El framework automatiza esta tarea
  - Se presentan complejidades para cuando se transfieren datos en objetos más complejos
    - Objetos de tipo “Map” y “List”
  - Es necesario implementar la conversión de objetos
- OGNL
  - Son las expresiones usadas para referenciar datos desde distintas zonas del framework
  - También se utiliza para obtener los datos desde el framework y usarlo en la presentación al usuario



- Transferencia de datos y conversión de tipos
  - Tarea común en aplicaciones web: convertir datos desde texto HTTP a objetos Java
  - Tarea tediosa y complicada
    - Llevar los datos desde un formulario a objetos Java
    - Leer Strings para construir Double, Floats, capturar exceptions por datos mal ingresados, etc
- La conversión de tipos se realiza en dos lados
  - Desde el requerimiento HTTP basado en texto a los JavaBeans
  - Cuando se muestra el resultado se convierte desde JavaBeans a HTML
    - Los datos se convierten desde tipos Java a texto



- OGNL: Object-Graph Navigation Language
  - Se integró en Struts2 para ayudar con la transferencia de datos y la conversión de tipos

```
<h5>Congratulations! You have created </h5>
```

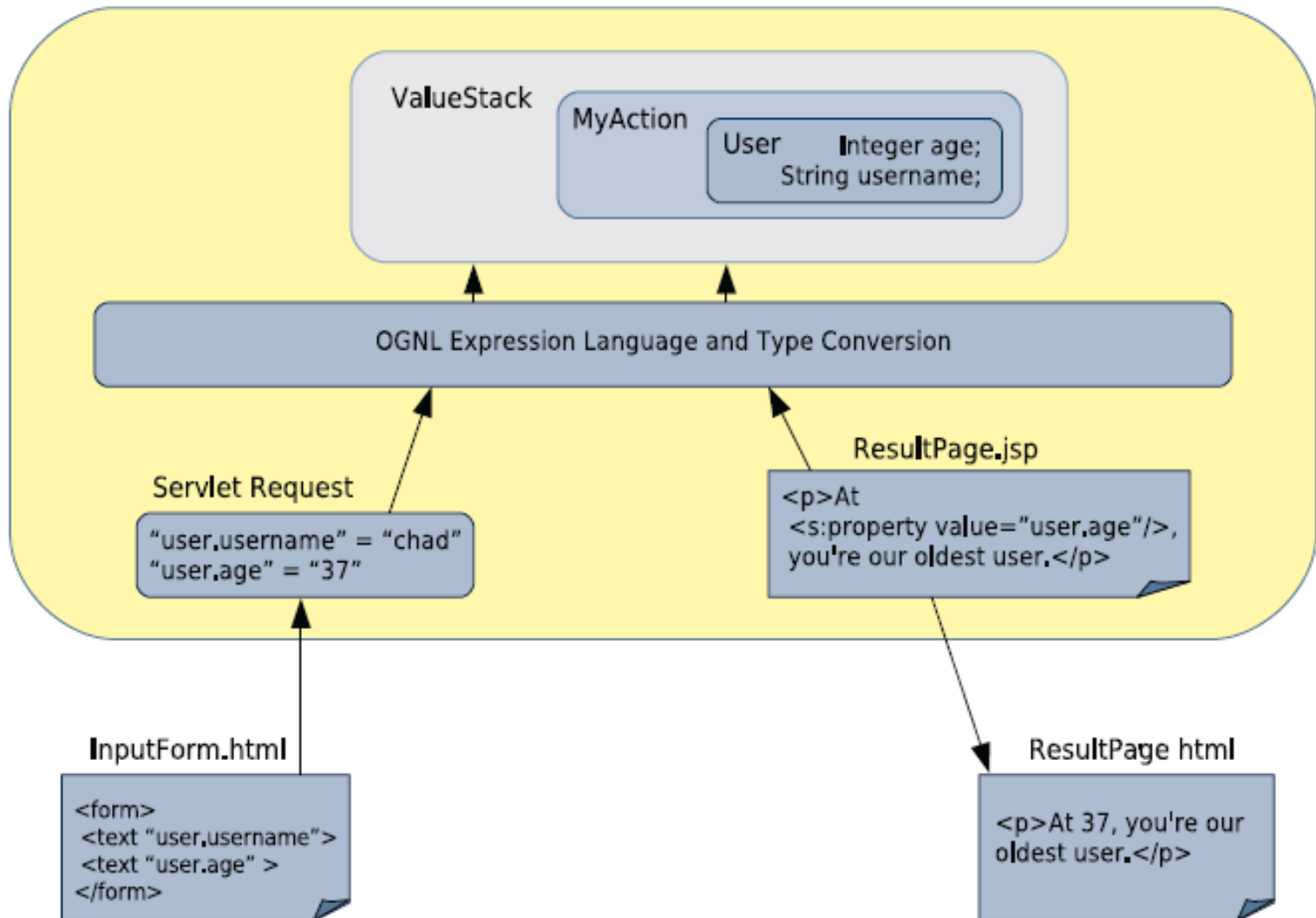
```
<h3>The <s:property value="portfolioName" /> Portfolio</h3>
```

- OGNL se está usando en el atributo value
  - Toma el valor desde una propiedad de los objetos Java y lo escribe en el HTML en lugar del TAG
  - Tag property vuelve al ambiente Java para obtener el valor del atributo “portfolioName”
  - Convierte desde el tipo de dato Java al formato de String en HTML

# OGNL



## Struts 2 Framework





- Asociando nombres de elementos en formulario con expresiones OGNL

```
<h4>Complete and submit the form to create your own  
portfolio.</h4>  
<s:form action="Register">  
<s:textfield name="user.username" label="Username"/>  
<s:password name="user.password" label="Password"/>  
<s:textfield name="user.portfolioName" label="Enter a name "/>  
<s:textfield name="user.age" label="Enter your age as a double "/>  
<s:textfield name="user.birthday" label="Enter birthday.  
(mm/dd/yy)"/>  
<s:submit/>  
</s:form>]
```

- Cada nombre de elemento en el formulario es una expresión OGNL



- Asociando nombres de elementos en formulario con expresiones OGNL
  - El action solo expone un grupo de propiedades de JavaBean que permiten transportar los datos
  - El requerimiento llega al framework con un “map” de pares nombre/valor
    - El nombre es una expresión OGNL
    - Se utiliza para buscar la propiedad correspondiente en el ValueStack
  - Se gatilla una “conversion exception” para cuando no se pueden asignar los valores
    - Se genera cuando se quiere llevar desde un String HTTP a un tipo de dato Java
    - Esto sucede antes de la ejecución del método validate()



- Parámetros multivalores de requerimiento HTTP
  - Se pueden convertir los valores a arreglos o “Collections”
  - Struts2 provee mecanismos para convertir datos en arreglos Java

```
<s:form action="ArraysDataTransferTest">
  <s:textfield name="ages" label="Ages"/>
  <s:textfield name="ages" label="Ages"/>
  <s:textfield name="ages" label="Ages"/>

  <s:textfield name="names[0]" label="names"/>
  <s:textfield name="names[1]" label="names"/>
  <s:textfield name="names[2]" label="names"/>

  <s:submit/>
</s:form>
```

**1** These target the ages property

**2** These target the names property





- Viajan 4 valores en la solicitud HTTP
  - Los 3 primeros tienen el mismo nombre, el parámetro viaja con 3 valores
  - El segundo grupo viajará cada uno en un parámetro distinto

Parameter name	Parameter value(s)
ages	12, 33, 102
names [0]	Chad
names [1]	Don
names [2]	Beth



- Listas

- A veces es necesario definir el tipo de datos que tendrá una lista (no siempre son String)
- Es necesario especificar el tipo de elemento que queremos para esa propiedad
  - OGNL utiliza archivos de propiedades para definir estos tipos de datos

