WOGenericContainer

Introduction

WOGenericContainer supports development of reusable components that closely model the behavior of common HTML elements. For example, along with WOComponentContent, you can use WOGenericContainer to implement your own hyperlink element as a reusable component. WOGenericContainer has attributes that support the takeValuesFromRequest and invokeAction phases of the component-action request-response loop.

Usage

```
WOGenericContainer {
  elementName = aConstantString;**
  [omitTags=aBoolean;]
  [elementID=identifier;]
  [otherTagString=aString;]
  [formValue=singleValue;]
  [formValues=arrayOfValues;]
  [invokeAction=aMethod;]... }
```

Bindings

elementName	Name of the HTML tag. This name (for example "textarea") will be used to generate the container's opening and closing tags (<textarea>...</textarea>). elementName can either be a constant or a variable, such as a key path. You can also set the value of this attribute to null, which effectively shuts off this element (that is, WebObjects doesn't generate HTML tags for this element). Alternatively, you can use the omitTags attribute to achieve the same effect. **This tag is actually optional, but because you want it 99.9% of the time, WOLips flags it as a validation error if it is missing. When would you omit the elementName? One example is the WOCheckboxMatrix component. A nameless WOGenericContainer is used as an invisible element which supplies the form name for the checkbox elements in the grid. If you need to use WOGenericContainer in this way, you can use a WOSwitchComponent to get around the validation error in your component.
omitTags	Specifies whether the element's tags should be displayed. This attribute is useful for defining an element that conditionally wraps HTML in a container tag. The default value is false. If omitTags is true, the contents of the tag are rendered but not the tags themselves. Using omitTags for a container makes the container itself optional.
elementID	Allows programmatic access to the element's element ID. This is a read-only attribute.
otherTagString	Enables any string to be part of the opening tag. This permits standalone attributes such as "checked" or "selected" to be part of a tag.
formValue, formValues	Enables implementation of input-type elements (for example, WOTextField). Bind these attributes to a variable that can contain the component's input value. During the takeValuesFromRequest phase, if the element ID of the current generic container matches an element ID of a form value in the request, the form value is pushed into the component using this attribute. The formValue attribute corresponds to WORequest's formValueForKey while the formValues attribute corresponds to WORequest's formValuesForKey method; in other words, formValue pushes a single attribute while formValues pushes an array of attributes.

invokeAction	Enables implementation of action elements (for example, WOHyperlink). During the invokeAction phase, if the element ID of the current generic container matches the sender ID of the URL, the method bound to this attribute is evaluated. Just as with any action method, it must return an object that conforms to the WOActionResults interface, such as WOComponent or WOResponse.

Examples

Java methods

WOD-style

```
```

Inline bindings (WOOGNL)

```
```

Related documents