

Cocoon Forms: submit widget (2.1 legacy document)

Table of contents

1 Comments.....3

Table of contents

1 Concept.....	3
2 Configuration.....	3

Warning:

This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Concept

The submit widget, usually rendered as a button, is used by the user to submit the form. The submit widget is a special kind of action widget, thus also has the same functionality as an action widget, however the submit widget does trigger validation and its purpose is to end the form.

You don't need to use a submit widget to submit a form. For example, in your HTML template you can simply put:

```
<input type="submit"/>
```

to be able to submit the form. Using a submit widget enables some extra functionality. You can control whether validation should be performed. You can put multiple submit widgets on a form and add different event handlers to them.

While a submit widget has the explicit purpose to submit a form, a form can also be submitted by other widgets. One example is the already mentioned [action widget](#). It is however also possible to automatically submit a form when a widget changes its value, so that server-side event value-changed event listeners can be triggered. This can be specified using `<fi:styling submit-on-change="true"/>` in the form template (see [XSLT](#) for more information on the `fi:styling` directive).

To know which widget caused the form to be submitted, use the method `getSubmitWidget` of the `Form` object.

2. Configuration

```
<fd:submit id="..." action-command="..." validate="true|false"> <fd:label>...</fd:label>  
<fd:help>...</fd:help> <fd:hint>...</fd:hint> <fd:on-action> ... </fd:on-action> </fd:submit>
```

The optional attribute `validate`, which is `true` by default, can be used to disable validation. The difference between an action widget and a submit widget with `validate="false"` is that a submit widget with `validate="false"` will end form processing, thus the form will not be redisplayed. Ultimately, it is of course the controller who decides this, but the forms hint towards the controller is that it shouldn't be redisplayed, and this is exactly what the flowscript integration library does.

1. Comments

add your comments