

HtmlCommandButtonAjax

Overview

Sends a form through ajax.

Usage

Build a form like normal, but use `x:commandButtonAjax` instead of `s:commandButton` (or standard `h:commandButton`)

```
<h:form>
  <h:outputText styleClass="standard_bold" value="Form Submit via AJAX"/>
  <h:panelGrid columns="2">
    <h:panelGrid columns="2">
      <t:inputText id="formText1" value="#{inputAjaxBean.formText1}"
        validator="#{inputAjaxBean.validateFormText1}"></t:inputText>
      <t:message for="formText1" forceSpan="true" styleClass="errorMessage"/>

      <t:inputText id="formText2" value="#{inputAjaxBean.formText2}"
        validator="#{inputAjaxBean.validateFormText2}"></t:inputText>
      <t:message for="formText2" forceSpan="true" styleClass="errorMessage"/>

      <s:commandButtonAjax action="#{inputAjaxBean.ajaxFormSubmit}" value="Submit"/>
      <h:outputText value="&nbsp;" escape="false"/>
    </h:panelGrid>
    <f:verbatim>This component demonstrates submitting a form through ajax.</f:verbatim>
  </h:panelGrid>
</h:form>
```

- `onSuccess` - called when an update is completely successful.
- `onFailure` - called when an update fails
- `onStart` - called when the update starts

Note: Do not include the brackets in these function callbacks.

Note 2: These functions accept two values (`elementName`, `elementValue`), for example:

```
function successful(ename, elvalue)
{
    // do something here
}
```

Note - This component is different to others, it uses Ajax mechanisms to fetch data from the server, thus the backend binding can and will be used outside of the standard JSF lifecycle to deliver the data.

Use always within a form tag.

see `examples/sandbox/inputAjax.jsp` for an example.