



Java™ Education & Technology Services

AJAX

JSF 2.0

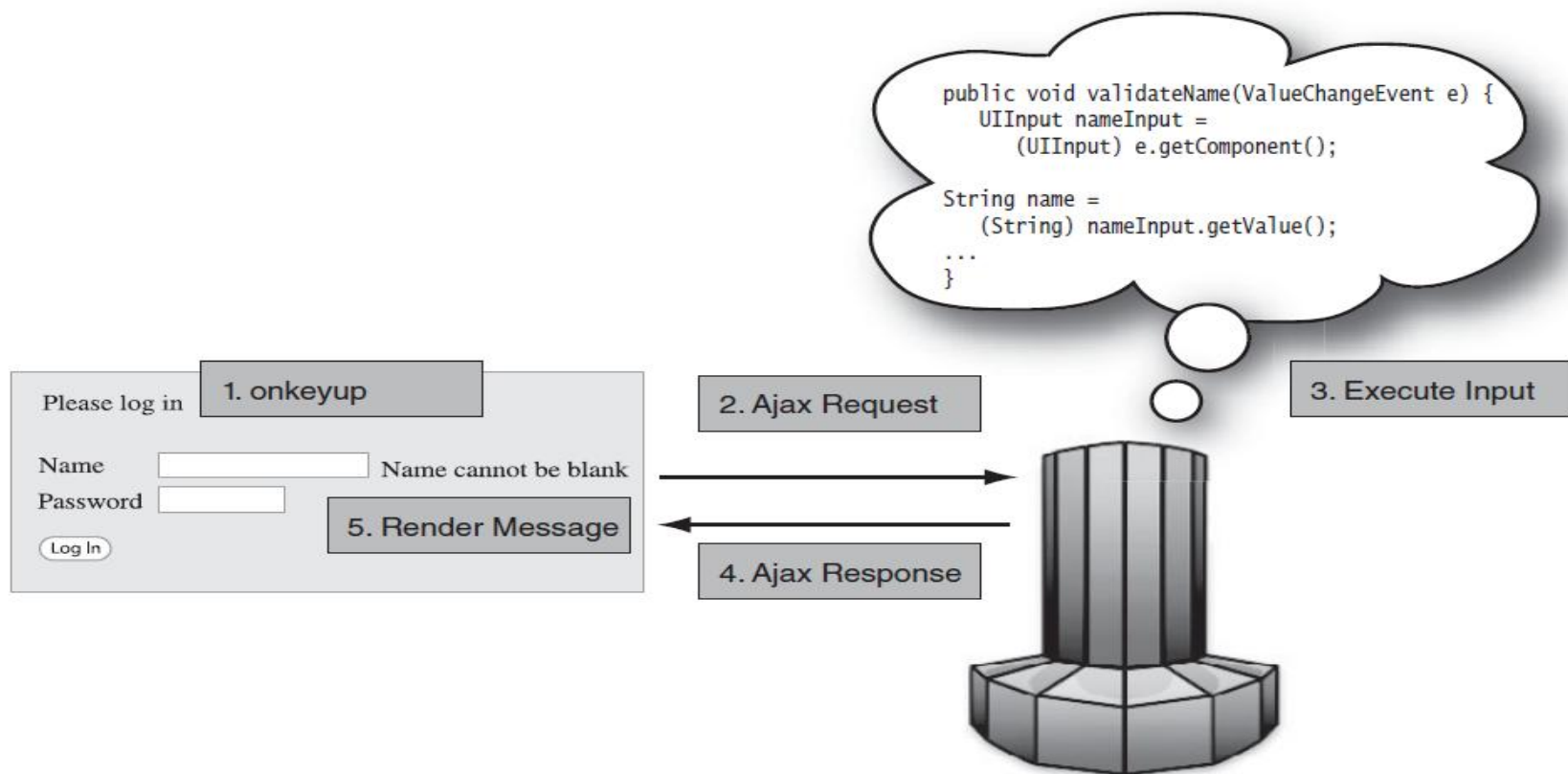


Chapter Outline

- ❑ Ajax and JSF
- ❑ f:ajax
- ❑ Ajax Request Monitoring

Ajax and JSF

1. Ajax partially processes forms on the server during the Ajax call.
2. Ajax partially renders Document Object Model (DOM) elements on





Ajax and JSF (cont'd)

- There are tons of Ajax libraries already (jQuery, DWR, GWT, etc.)

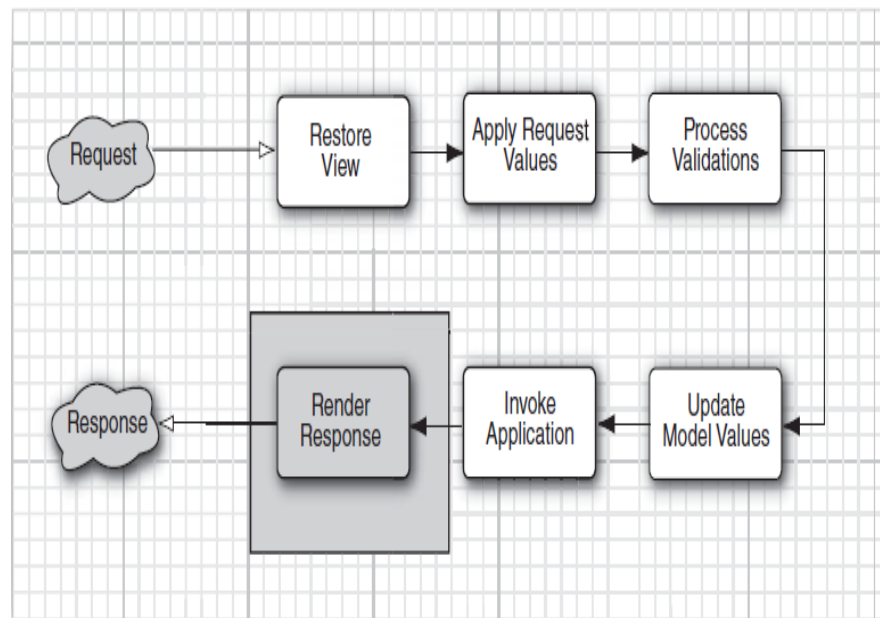
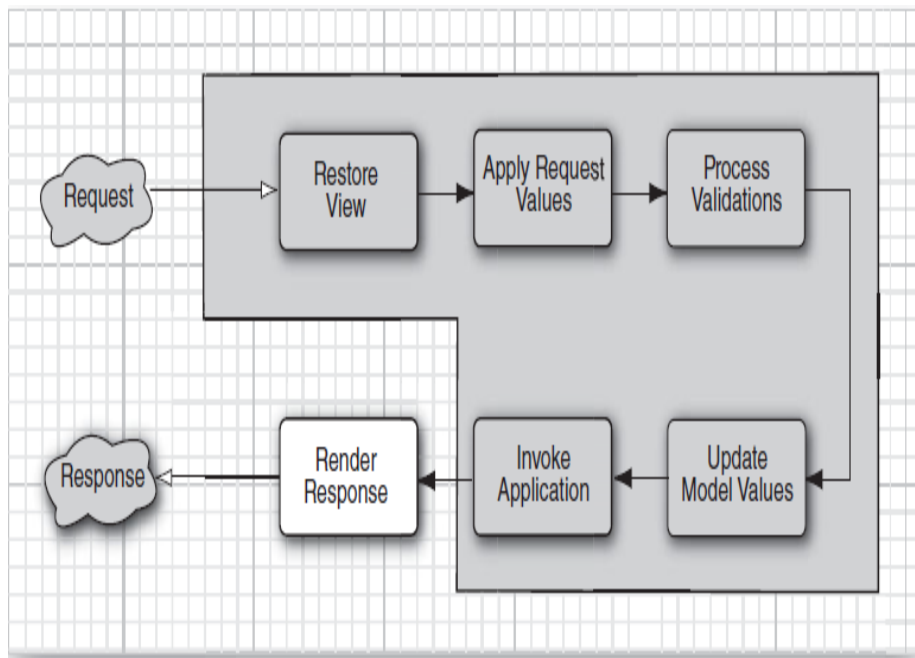
Why a JSF-specific Ajax library?

- Client side
 - You can update JSF **elements** (h:outputText, h:inputText, h:selectOneMenu, etc.)
 - You **don't** have to write **JavaScript**
- Server side
 - Ajax calls know about JSF **managed beans** including reading form fields and setting bean properties
 - You don't have to write **servlets** and parse **parameters**

The JSF Life Cycle and Ajax

JSF 2.0 splits the JSF life cycle into two parts:

- **execute** :you specify a set of components that JSF executes
- **render**: you specify a set of components that it renders.



The JSF Life Cycle and Ajax

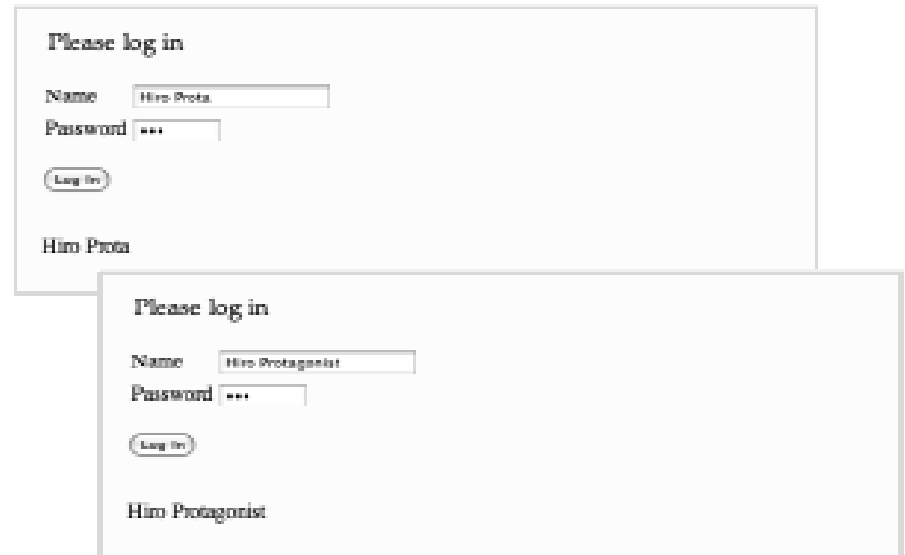
1. Converts and validates the **execute** component's value (if the component is an input)
2. Pushes valid input values to the model (if the component is wired to a bean property)
3. Subsequently renders the **render** components.

```
<h:inputText id="name" value="#{user.name}">
```

```
<f:ajax event="keyup" execute="@this" render="echo"/>
```

```
</h:inputText>
```

```
<h:outputText id="echo" value="#{user.name}"/>
```



The image displays two overlapping screenshots of a web login form to demonstrate an Ajax update. The top form, representing the initial state, has the title "Please log in", a "Name" field with the value "Hiro Prota", a "Password" field with masked characters "***", a "Log In" button, and a feedback message "Hiro Prota". The bottom form, representing the state after an Ajax update, has the same title and fields, but the "Name" field now contains "Hiro Protagonist" and the feedback message has updated to "Hiro Protagonist".

f:ajax Tag Attributes:

- **event:**

The DOM event without the on prefix (event="blur")
can be the component events action and valueChange

- **execute:**

A space-separated list of components that JSF executes on the server during the Ajax call.

Valid keywords:

| | |
|------------------|-------|
| @this (default) | @form |
| @all | @none |

- **render:**

A space-separated list of components that JSF renders on the client after the Ajax call returns from the server.

f:ajax Tag Attributes:

- **onevent :**

A JavaScript function that JSF calls for Ajax events.

will be called three times during the lifetime of a successful Ajax call:

- begin (Ajax call begins)
- complete (has been processed on the server)
- success (just before JSF renders the Ajax response)

- **onerror:**

A JavaScript function that JSF calls if the Ajax call results in an error.

Examples

```
<f:ajax render="...">
    <h:form>
        <h:panelGrid columns="2">
            ...
            <h:inputText id="name" value="#{user.name}"/>
            ...
            <h:inputText id="password" value="#{user.password}"/>
            ...
            </h:commandButton value="Submit" action="#{user.login}"/>
        </h:panelGrid>
    </h:form>
</f:ajax>
```

Examples

```
<f:ajax render=" "      event="click">
  <h:form>
    <h:panelGrid columns="2">
      ...
      <h:inputText id="name" value="#{user.name}"/>
      ...
      <h:inputText id="password" value="#{user.password}"/>
      ...
      </h:commandButton value="Submit" action="#{user.login}"/>
    </h:panelGrid>
  </h:form>
</f:ajax>
```

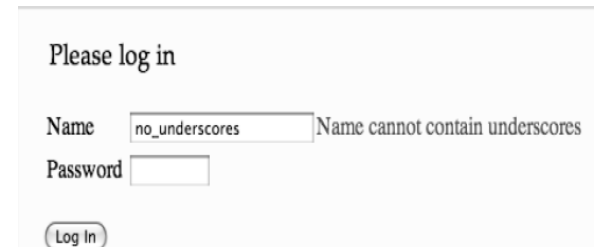
Examples (Ajax Field Validation)

```
<h:inputText id="name" value="#{user.name}"
    validator= "#{user.validateName}" >
```

```
    <f:ajax event="blur" render="nameError"/>
</h:inputText>
<h:message for="name" id="nameError" style="color: red" />
```

// In Bean

```
public void validateName(FacesContext fc, UIComponent c, Object value) {
    if ( ((String)value).contains("_") )
        throw new ValidatorException(new FacesMessage("Name cannot contain underscores"));
}
```



Please log in

Name Name cannot contain underscores

Password



Ajax Request Monitoring

<f:ajax render="..." **onevent**="functionName" ... />

<script type="text/javascript">

function **functionName**(data) {

.....

}

| Attribute | Description |
|--------------|--|
| status | The status of the current Ajax call. Must be one of: begin, complete, or error. |
| source | The DOM element that is the source of the event. |
| responseXML | The response to the Ajax request. This object is undefined in the begin phase of the Ajax request. |
| responseText | The XML response, as text. This object is also undefined in the begin phase of the Ajax request. |

Ajax Request Monitoring (cont'd)

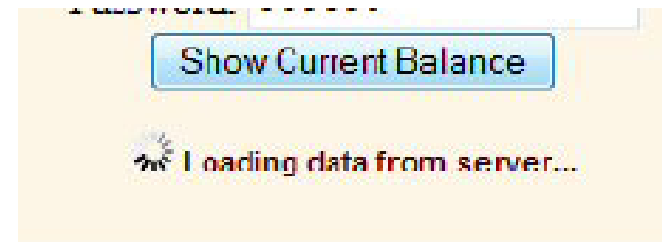
Example

- **Idea**

- You have slow server operation
- Display animated GIF (& message) when request sent
- Hide GIF/message when response completes

- **Approach**

- Get animated GIF
- Display image plus message in region with display: none So it is hidden initially
- When request begins, change to display: inline
 - Use onevent handler and “begin” status
- When request finishes successfully, make display: none
 - Use onevent handler and “success” status





Ajax Request Monitoring (cont'd)

Example

```
<h:commandButton >
```

```
    <f:ajax execute="@form" render="ajaxMessage2"  
        onevent="showWorkingIndicator"/>
```

```
</h:commandButton>
```

```
    :  
    :
```

```
<span id="workingIndicator" style="display: none; font-size: 60%">
```

```
    
```

Loading data from server...

```
</span>
```

Example (java script)

```
function showWorkingIndicator(data) {
    showIndicatorRegion(data, "workingIndicator");
}

function showIndicatorRegion(data, regionId) {
    if ( data.status == "begin" ) {
        showElement(regionId);
    } else if (data.status == "success") {
        hideElement(regionId);
    }
}

function showElement(id) {
    document.getElementById(id).style.display = "inline";
}

function hideElement(id) {
    document.getElementById(id).style.display = "none";
}
```




Lab Exercise

- Ajax Field Validation

Please log in

Name Name cannot contain underscores

Password