

JSF Navigation, AJAX and Component Suites

JSF Slides #4

JSF Navigation Model

Implicit navigation

- In pages or Java code

Rule based navigation

- In **faces-config.xml**
- Variations
 - Static
 - Dynamic
 - Conditional

Remember: Late in request cycle

Implicit Navigation

Strings interpreted as filename (no xhtml suffix)

```
<!-- Navigate to page confirm.xhtml -->
<h:commandButton ... action="confirm" ../>

<!-- Navigate to outcom of method -->
<h:commandButton ... action="#{bean.navigate}" ../>
```

In bean

```
public String navigate(){
    ..
    return "aFileNameWithoutSuffix";
}
```

Rule Based Navigation

Rules applied in written order, first success wins

```
<!-- faces-config.xml -->
<navigation-rule>
  <from-view-id>/viewProducts.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>onViewCart</from-outcome>
    <to-view-id>/viewCart.xhtml</to-view-id>
  </navigation-case>
  <!-- More cases -->
</navigation-rule>
```

Variations

- Static, hard coded
- Dynamic, from bean method result
- Conditional, possible to avoid "first success wins"

Redirects

Possible to do redirect using a predefined string
- As usual parameters don't survive

```
<h:commanLink ...action="to?faces-redirect=true".../>
```

Or a tag in faces-config.xml

```
<navigation-rule>  
  <from-view-id>/searchForm.jsp</from-view-id>  
  <navigation-case>  
    <from-outcome>submit</from-outcome>  
    <to-view-id>/searchResults.jsp</to-view-id>  
    <redirect/>  
  </navigation-case>  
</navigation-rule>
```

JSF PRG pattern

PRG pattern (remainder: post-redirect-get)

- Possible, use flash object or view parameters ...
- ... but probably simplest with `@ConversationalScope` (for CDI managed beans)

See code samples

JSF and AJAX

JSF is default “full page load”

- Suitable for many application

Possible to add AJAX capability

- Real need ...??

Two ways to add AJAX behaviour

- Using <f:ajax> tag, **declarative** AJAX, we avoid...!
- Using raw Javascript (we don't). The standard Javascript resource library bundled with JSF (jsf.js). Global jsf-object, **programmatic** AJAX, ... we don't
- ... better use a higher level component suite... upcoming

JSF AJAX tag

<f:ajax> will add AJAX behavior to component(s)

- Nested inside component tag or..
- Enclose component tag(s)
- Both possible (behavior: Union of all <f:ajax>)

Many attributes defining

- Which component (id) AJAX applies to (execute)
- What event (onclick, ...) (event)
- What to render (insert) result (render)
- Others: listener, immediate, disabled

JSF AJAX tag cont.

Some special keyword to use for "execute" and "render" attributes of <f:ajax> (there are more)

execute= "@..."

- @all, every component is submitted/processed
- @form, submit/process entire form

render = "@..."

- @all, all components rendered
- @form, render entire form

JSF Resources

For CSS, images, JavaScript, ...using the "library" attribute

Example: NetBeans project/Maven

CSS in Web Pages/resources/**css**, JavaScript in Web Pages/resources/**js**, etc

```
<html>
  <h:outputStylesheet library="css" name="styles.css" />
  <h:outputScript library="js" name="utils.js" target="head"/>
  <body>
    <h:graphicImage library="img"
      name="tomato.jpeg" alt="tomato"/>
```

JSF I18N

Internationalization (i18n) using resource bundles

- msg.properties, msg_de.properties, msg_sv.properties, ...
- Flat text files to Map<String, String>

In faces-config.xml

```
<resource-bundle>  
    <!-- Package and folder hierarchy (must be nested) -->  
    <base-name>edu.ch1.hajo.i18n.msg</base-name>  
    <!-- This is the name used in EL-expressions -->  
    <var>msg</var>  
</resource-bundle>
```

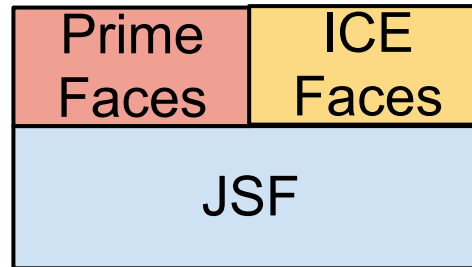
In page

```
<..."&#106;{msg.lblWelcome}".../>
```

Higher Level JSF Component Suites

Many higher level component suites built on top of JSF

- ICEFaces
- PrimeFaces
- ...



Higher level components

- Calendars, Datatables (editable, sortable,...), many more...

ICEFaces

Dependency in pom.xml

- See samples

Two different component suites: <ice: .../> and <ace: .../>

- **We use ace** (latest, greatest)

```
<!-- Using ICEFaces tags in page -->
<html xmlns:ace="http://www.icefaces.org/icefaces/components"
.../>
```

ICEFaces Datatable

```
<h:form id="personForm">
  <ace:dataTable id="personTableSingleRow"
    value="#{persons.data}" var="p" rows="3" paginator="true"
    paginatorPosition="bottom" selectionMode="single"
    resizableColumns="true"
    stateMap="#{persons.stateMap}"
    rowSelectListener="#{persons.selectListener}">
    <ace:column id="id" headerText="Id" sortBy="#{p.id}"> #{p.id}
    </ace:column>
    <ace:column id="name" headerText="Name">
      #{p.name}
    </ace:column>
    <ace:column id="age" headerText="Age">
      #{p.age}
    </ace:column>
  </ace:dataTable>
  <h:outputText value="#{persons.selected}" /><br/>
</h:form>
```