Jsf;

Ajax,jsf集成框架，ajax4jsf；

JBoss 收购 A4J 后，又将其集成到 RichFaces 中RichFaces；---公司4.35版；

~

ajax4jsf，无刷新操作实现；

最新版本：5.0.x；

参考：

RichFaces官网文档，

<http://docs.jboss.org/richfaces/latest_4_3_X/Component_Reference/en-US/html_single/>

示例，

<http://showcase.richfaces.org/>

-----------------------------

目录：

功能；

控件使用；

//////////////////////////////////////////////////////////////////////////////////////

Xhtml：

Richfaces标签库引入每个xhtml页面，  
<ui:composition xmlns:r="http://richfaces.org">

……

</ui:composition>

功能：

自定义皮肤：

///////////////////////////////////////////////////////////////////////////

控件使用：

主要控件：

ajax按钮； <a4j:commandButton>

弹出窗体；<rich:popupPanel> 浮层面板；

Ajax：

1，<a4j:ajax>

<a4j:ajax event="keyup" render="outtext" />

2，<a4j:param> ---加入参数；

<h:form **id**="form">

    <a4j:commandButton **value**="Set name to Alex" **reRender**="rep">

        <a4j:param **name**="username" **value**="Alex" **assignTo**="#{paramBean.name}"/>

    </a4j:commandButton>

    <h:outputText **id**="rep" **value**="Name: #{paramBean.name}"/>

</h:form>

点击按钮，设置rep控件中的的me值为alex；

3.3. <a4j:actionListener>

3.4. <a4j:commandButton>

Id,name,value，

Onclick，---action提交前执行，一般为js事件；

Action,

Execute,---@form,提交时携带表单信息；@this,提交时不携带信息；

render,---控件指向，

data,--- data="#{borrowMoneyAction.resultMsg}"，action返回提示信息；

oncomplete 在客户端请求完成时调用的js代码;

3.5. <a4j:commandLink>

Id,name,value,

onclick；

3.6. <a4j:jsFunction>

<h:outputText id="showname" value="#{functionBean.text}" />

<h:form>

<a4j:jsFunction name="updateName" render="showname">

<a4j:param name="name" assignTo="#{functionBean.text}"/>

</a4j:jsFunction>

</h:form>

鼠标移走后，改变内容；

3.7. <a4j:poll>

3.8. <a4j:push>

4.1. <a4j:mediaOutput> ---资源，

<a4j:mediaOutput element="img" cacheable="false" session="false"

createContent="#{mediaBean.paint}" value="#{mediaData}"

mimeType="image/jpeg" />

可以生成一张图片；

5.1. <a4j:outputPanel>

5.2. <a4j:region>

7.1. <a4j:queue>

<a4j:queue requestDelay="1500"/> 队列请求延迟1500毫秒；

7.2. <a4j:log>

7.3. <a4j:status>

---------------------

execute ：

@All

每个组件被处理。

@none

没有部件被处理。

@This

与execute属性一起的请求组件被处理。

@form

form包含的请求组件被处理。

@region

region包含的请求组件被处理。使用<R：region>组件作为包装元素来指定区域。有些组件利用额外的关键字。

//////////////////////////////////////////////////////////////////////////////////

6.1. <rich:validator> ---验证，

<rich:panel header="User information">

<h:panelGrid columns="3">

<h:outputText value="Name:" />

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

<rich:validator />

</h:inputText>

<rich:message for="name" />

<h:outputText value="Email" />

<h:inputText value="#{validationBean.email}" id="email">

<rich:validator />

</h:inputText>

<rich:message for="email" />

<h:outputText value="Age" />

<h:inputText value="#{validationBean.age}" id="age">

<rich:validator />

</h:inputText>

<rich:message for="age" />

<h:outputText value="I agree the terms" />

<h:selectBooleanCheckbox value="#{validationBean.agree}" id="agree">

<rich:validator/>

</h:selectBooleanCheckbox>

<rich:message for="agree" />

</h:panelGrid>

</rich:panel>

@ManagedBean

@RequestScoped

public class UserBean {

@Size(min=3, max=12)

private String name = null;

@Pattern(regexp = "^[\\w\\-]([\\.\\w])+[\\w]+@([\\w\\-]+\\.)+[a-zA-Z]{2,4}$" , message="Bad email")

private String email = null;

@Min(value = 18)

@Max(value = 99)

private Integer age;

//...

//Getters and Setters

}

<rich:graphValidator>

·示例代码；---可用于登录页面验证；

--------------------------

8. Rich inputs

8.1. <rich:autocomplete> 自动补全；

8.2. <rich:calendar> 日期控件；

8.3. <rich:editor> 富文本；

8.4. <rich:fileUpload> 上传；

8.5. <rich:inplaceInput> 文本框；

8.6. <rich:inplaceSelect> 下拉框；---有一个集合；

8.7. <rich:inputNumberSlider> 数字滑块；

8.8. <rich:inputNumberSpinner> 数字微调；

8.9. <rich:select> 下拉框；

8.10. <rich:orderingList> 列表；

8.11. <rich:pickList> 左右互选列表；

9. Panels

9.1. <rich:panel> 面板

9.2. <rich:accordion>

9.3. <rich:collapsiblePanel>

9.4. <rich:popupPanel> 浮层面板；

9.5. <rich:tabPanel> 选项卡；

9.6. <rich:togglePanel>

10. Tables and grids

10.1. <a4j:repeat> 表格数据循环---加载list数据；

10.2. <rich:dataTable> 表格；

10.3. <rich:column> 列；

10.4. <rich:columnGroup> 合并列；

10.5. <rich:collapsibleSubTable>

10.6. <rich:extendedDataTable> 可扩展表格(滚动条)

10.7. <rich:dataGrid> 网格表；

10.8. <rich:list> 列数据；

10.9. <rich:dataScroller> 滚轮表格；

10.10. Table filtering

11. Trees

11.1. <rich:tree> 树；

11.2. Tree adaptors

12，Menus and toolbars

12.1. <rich:dropDownMenu> 下拉菜单；

12.2. <rich:contextMenu>

12.3. Menu sub-components

12.3.1. <rich:menuItem>

12.3.2. <rich:menuGroup>

12.3.3. <rich:menuSeparator>

12.4. <rich:panelMenu> 面板菜单---可用于后台左侧菜单；

12.5. <rich:toolbar> 工具栏；

13. Output and messages

13.1. <rich:message> 输出信息到output控件；

13.2. <rich:messages>

13.3. <rich:notify>

13.4. <rich:notifyMessage>

13.5. <rich:notifyMessages>

13.6. <rich:notifyStack>

13.7. <rich:progressBar> 进度条；

13.8. <rich:tooltip> 提示；

14. Drag and drop

14.1. <rich:dragSource>

14.2. <rich:dropTarget>

14.3. <rich:dragIndicator>

15. Layout and appearance

15.1. <rich:jQuery>

16. Functions

16.1. rich:clientId

16.2. rich:component

16.3. rich:element

16.4. rich:jQuery

16.5. rich:findComponent

16.6. rich:isUserInRole

17. Functionality extension

17.1. <rich:componentControl>

17.2. <rich:focus>

17.3. <rich:hotKey>

17.4. <rich:hashParam>

17.5. <rich:placeholder>

/////////////////////////////////////////////////////////////////////////////

<rich:popupPanel*弹出窗体；*

 <h:commandButton value=*"Call the popup"*>

<rich:componentControl target=*"popup"* operation=*"show"* />

</h:commandButton>

<rich:popupPanel id=*"popup"* modal=*"false"* autosized=*"true"* resizeable=*"false"*>

<f:facet name=*"header"*>

<h:outputText value=*"Simple popup panel"* />

</f:facet>

<f:facet name=*"controls"*>

<h:outputLink value=*"#"* onclick="#{rich:component('popup')}.hide(); return false;">

X

</h:outputLink>

</f:facet>

<p>Any content might be inside this panel.</p>

<p>

The popup panel is open and closed from the javascript function of component client side object. The following code

<a href=*"#"* onclick="#{rich:component('popup')}.hide()">hide this panel</a>:

<f:verbatim>&#35;</f:verbatim>{rich:component('popup')}.hide()

</p>

</rich:popupPanel>