Jsf：视图层实现框架；| 类似，struts2；

Java7中版本2.2；

参考：

Ibm三篇文章，

<http://www.ibm.com/developerworks/cn/java/j-jsf2fu1/index.html>

官方文档，java7文档中，

<http://docs.oracle.com/javaee/>

7，8，9，

10，jsf控件：html标签，f标签；

11-转换，监听，验证；

12，13，14，15，16；

api：

<https://javaserverfaces.java.net/nonav/docs/2.1/>

或上面java7文档中的；

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

.xhtml页面；

主内容：

页面；

配置文件；

业务逻辑中的相关；

Facelets-模版框架，

复合组件；

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

目录：

页面控件使用；

页面控件高级使用；

功能实现；

附录-控件；

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

页面控件使用：

页面赋值：

#{name}

基本控件1，ui:composition、c:set、h:ead、h:body；

<ui:composition xmlns=*"http://www.w3.org/1999/xhtml"*

xmlns:h=*"http://java.sun.com/jsf/html"*

xmlns:f=*"http://java.sun.com/jsf/core"*

xmlns:ui=*"http://java.sun.com/jsf/facelets"*

xmlns:a4j=*"http://richfaces.org/a4j"*

xmlns:rich=*"http://richfaces.org/rich"*

xmlns:shiro=*"http://xproject.flightCenter/tag"*

xmlns:c=*"http://java.sun.com/jsp/jstl/core"*>

<c:set var=*"path"* value=*"${request.contextPath}"* />

<h:head>

</h:head>

<h:body>

</h:body>

</ui:composition>

基本控件2，

<h:form prependId="false">

<h:panelGrid columns="2">

<h:inputText value="#{place.streetAddress}" size="15"/> 输入框

-id,name,value；

<h:outputText value="#{UserNumberBean.maximum}"/> 文本行；

<h:commandButton value="#{msgs.goButtonText}"

style="font-family:Palatino;font-style:italic"

action="#{place.fetch}"/>

-id,value,action；

<h:graphicImage library="images" name="cloudy.gif"/> 图片；

//图标组件，

<util:icon image="#{resource['images:back.jpg']}"

actionMethod="#{places.logout}"

style="border: thin solid lightBlue"/>

//密码

<h:inputSecret redisplay="false"

value="#{loginBean.password}" />

//输出内容为“hello,（f:param的value值）”

<h:outputFormat value="Hello, {0}!">

<f:param value="#{hello.name}"/>

</h:outputFormat>

//dataTable中取得集合数据，

<h:dataTable id="items"

<h:column>

<h:message 信息；

<h:link outcome="personal" value="personal greeting page!"

includeViewParams="true">

<f:param name="Result" value="#{hello.name}"/>

</h:link>

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

控件高级使用：

引入js和css外部文件，

<h:outputStylesheet library="css" name="styles.css" target="body"/>

<h:outputScript library="javascript" name="util.js" target="head"/>

##### Facelets模版：/templates/masterLayout.xhtml

<div class="pageHeading">

<**ui:insert** name="heading">

#{msgs.placesHeading}

</ui:insert>

</div>

<div class="menuLeft">

<**ui:insert** name="menuLeft"/>

</div>

<**ui:insert** name="menuRight">

<ui:include src="/sections/shared/sourceViewer.xhtml"/>

</ui:insert>

ui:define，覆盖内容；

login 视图

<ui:composition xmlns="http://www.w3.org/1999/xhtml"

xmlns:ui="http://java.sun.com/jsf/facelets"

template="/templates/masterLayout.xhtml">

<ui:define name="menuLeft">

<ui:include src="/sections/login/menuLeft.xhtml"/>

</ui:define>

<ui:define name="content">

<ui:include src="/sections/login/content.xhtml"/>

</ui:define>

</ui:composition>

~

也可以修改menuRight内容；

<ui:define name="menuRight">

<ui:include src="/sections/showSource/menuRight.xhtml"/>

</ui:define>

实现定制组件：---组件开发模式；

·定义组件，然后使用；| 示例：login组件；

·复合嵌套组件；

Ajax化：

Ajax验证，

<h:inputText id="name" value="#{cc.attrs.managedBean.name}"

**valueChangeListener**="#{cc.attrs.managedBean.validateName}">

<**f:ajax** event="**blur**" render="**nameError**"/>

</h:inputText>

<h:outputText id="**nameError**"

value="#{cc.attrs.managedBean.**nameError**}"

style="color: red;font-style: italic;"/>

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

转换：

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

<f:converter converterId="javax.faces.Integer" />

</h:inputText>

<h:outputText value="#{cashierBean.shipDate}">

<f:convertDateTime pattern="EEEEEEEE, MMM dd, yyyy" />

</h:outputText>

~

<f:convertDateTime type="date" dateStyle="full" />

~

<f:convertDateTime dateStyle="full"

locale="es"

timeStyle="long" type="both" />

<h:outputText value="#{cart.total}">

<f:convertNumber currencySymbol="$" type="currency"/>

</h:outputText>

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

功能实现：

**表 2. JSF 2 托管 bean 注释（@...Scoped 注释只对 @ManagedBean 有效）**

| **托管 bean 注释** | **描述** | **属性** |
| --- | --- | --- |
| @ManagedBean | 以托管 bean 的形式注册一个类实例，然后将其放入到使用其中一个 @...Scoped 注释指定的范围内。如果没有指定任何范围，JSF 将把此 bean 放入请求范围，如果没有指定任何名称，JSF 将把类名的第一个字母转换为小写，形成一个托管 bean 名称；例如，如果类名为UserBean，那么 JSF 将创建一个托管 bean，其名为 userBean。eager 和 name 属性都是可选的。  注释必须结合使用一个实现零参数构造器的 Java 类。 | eager，name |
| @ManagedProperty | 为托管 bean 设置一个属性。注释必须放在类成员变量的声明之前。name 属性指定特性的名称，默认情况下为成员变量的名称。value 属性是特性的值，可以是一个字符串，也可以是一个 JSF 表达式，比如 #{...}。 | value，name |
| @ApplicationScoped | 在应用程序范围内存储托管 bean。 |  |
| @SessionScoped | 在会话范围内存储托管 bean。 |  |
| @RequestScoped | 在请求范围内存储托管 bean。 |  |
| @ViewScoped | 在视图范围内存储托管 bean。 |  |
| @NoneScoped | 将托管 bean 指定为没有范围。无范围的托管 bean 在被其他 bean 引用时比较有用。 |  |
| @CustomScoped | 在定制范围内存储托管 bean。  定制范围就是指可以由页面创建者访问的地图。可以通过编程的方式控制定制范围内的 bean 的可视性和生命周期。value 属性指向一个地图。 | value |

项目业务逻辑类，

@ManagedBean(name = "productAction")

@ViewScoped

**public** **class** ProductAction **implements** Serializable{

//service类，

@ManagedProperty(name = "financialProductService", value = "#{financialProductService}")

**private** FinancialProductService financialProductService;

//实体类---理财产品对象

@ManagedProperty(name = "financialProduct", value = "#{financialProduct}")

**private** FinancialProduct financialProduct;

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

开发一个managed baen：

页面；

控件，按钮，取值，传值；

~

类，

@Named

@SessionScoped

public class UserNumberBean implements Serializable {

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

附录-控件：

Html标签，

| **Tag** | **Functions** | **Rendered As** | **Appearance** |
| --- | --- | --- | --- |
| h:column | 列 | A column of data in an HTML table | A column in a table |
| h:commandButton | 提交表单到应用程序 | An HTML <input type=*value*> element for which the type value can be "submit", "reset", or "image" | A button |
| h:commandLink | 链接到另一页面或本页一位置 | An HTML <a href>element | A link |
| h:dataTable | 数据表 | An HTML <table> element | A table that can be updated dynamically |
| h:form | 表单 | An HTML <form> element | No appearance |
| h:graphicImage | 显示图像 | An HTML <img> element | An image |
| h:inputFile | 上传文件控件 | An HTML <input type="file"> element | A field with a**Browse...** button |
| h:inputHidden | 隐藏控件 | An HTML <input type="hidden"> element | No appearance |
| h:inputSecret | 加密控件 | An HTML <input type="password"> element | A field that displays a row of characters instead of the actual string entered |
| h:inputText | 文本框 | An HTML <input type="text"> element | A field |
| h:inputTextarea | 文本框域 | An HTML <textarea>element | A multirow field |
| h:message | 显示一个本地消息 | An HTML <span> tag if styles are used | A text string |
| h:messages | 显示本地消息 | A set of HTML <span> tags if styles are used | A text string |
| h:outputFormat | 显示格式化信息 | Plain text | Plain text |
| h:outputLabel | 文本 | An HTML <label> element | Plain text |
| h:outputLink | 链接到页面上的其他页面或位置，而不会产生一个动作事件 | An HTML <a> element | A link |
| h:outputText | 行文本 | Plain text | Plain text |
| h:panelGrid | 表格 | An HTML <table> element with <tr> and <td>elements | A table |
| h:panelGroup | 一个组件下的表格 | A HTML <div> or <span>element | A row in a table |
| h:selectBooleanCheckbox | 选择框 | An HTML <input type="checkbox"> element | A check box |
| h:selectManyCheckbox | 多选框 | A set of HTML <input>elements of type checkbox | A group of check boxes |
| h:selectManyListbox | 允许用户选择一组项目的同时显示多个项目 | An HTML <select>element | A box |
| h:selectManyMenu | 允许用户从一组项目中选择多个项目 | An HTML <select>element | A menu |
| h:selectOneListbox | 允许用户从一组项目中选择一个项目一次性全部显示 | An HTML <select>element | A box |
| h:selectOneMenu | 允许用户从一组项目中选择一个项目一次性全部显示 | An HTML <select>element | A menu |
| h:selectOneRadio | 选择一个按钮 | An HTML <input type="radio"> element | A group of options |

Core标签：

| **Tag** | **Function** |
| --- | --- |
| f:actionListener | 添加一个动作监听父组件 |
| f:phaseListener | 添加一个PhaseListener到一个页面 |
| f:setPropertyActionListener | 注册一个特殊的动作监听器，其唯一目的是当表单被提交时，推动一个值到一个托管bean |
| f:valueChangeListener | 添加一个值变化，监听到父组件 |
| f:converter | 添加任意转换器到父组件 |
| f:convertDateTime | 添加一个DateTimeConverter实例到父组件 |
| f:convertNumber | 添加一个NumberConverter实例到父组件 |
| f:facet | 封闭标签的嵌套组件 |
| f:metadata | 注册一个方面到父组件 |
| f:selectItem | 代表项目列表中的一个项目 |
| f:selectItems | 代表一组项目 |
| f:attribute | 父组件的属性/添加属性到父组件 |
| f:loadBundle | Map到资源包 |
| f:param | 参数 |
| f:ajax | ajax |
| f:event | 允许在组件上安装ComponentSystemEventListener |