

# Java Server Faces (JSF)

Indy Java User Group

July 26, 2006

Matt DeCapua

[mjdecap@yahoo.com](mailto:mjdecap@yahoo.com)

# Outline

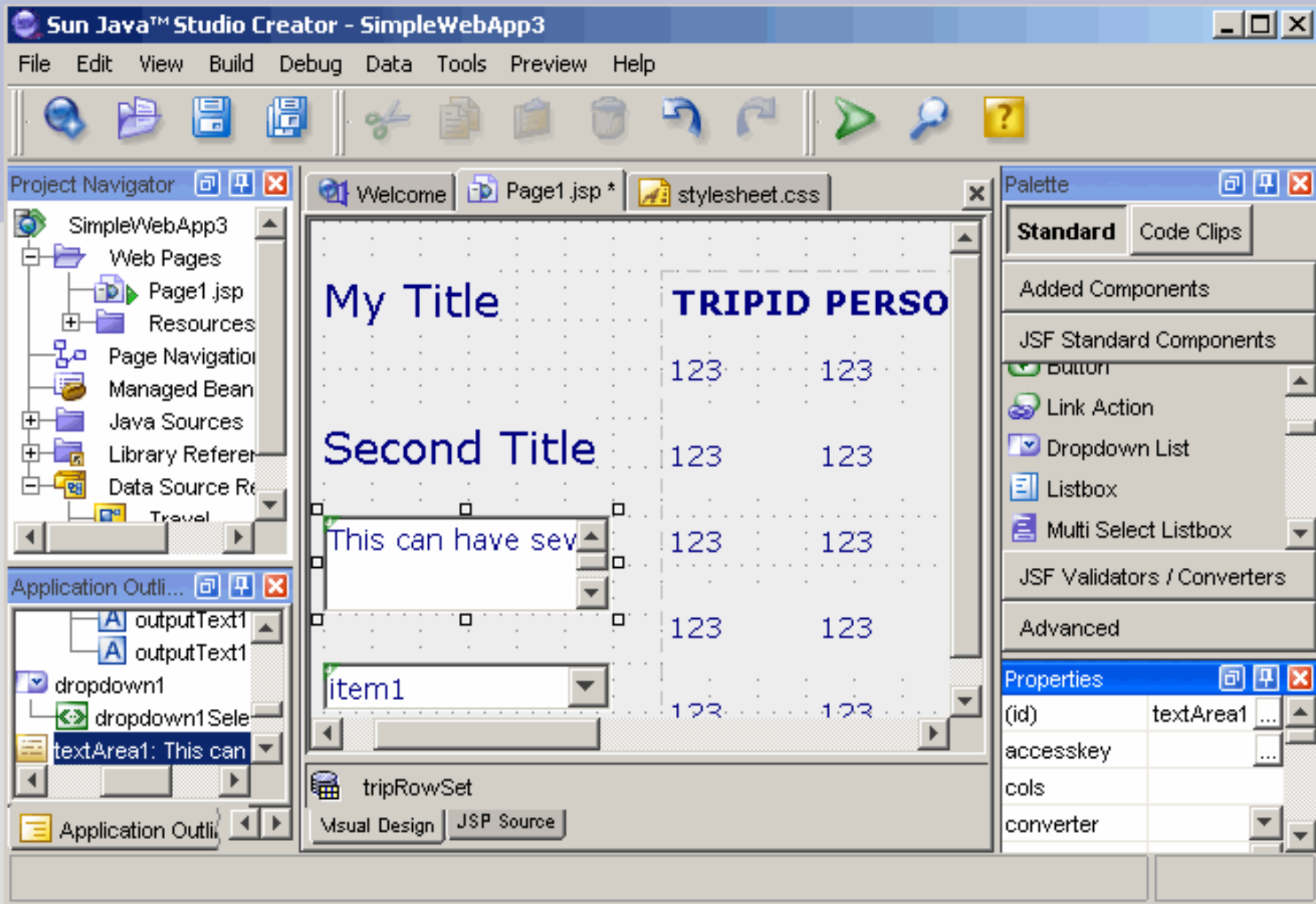
- What is JSF
- Brief History
- JSF Lifecycle
- Demo Specs
- Setting Up JSF
- Components
- Configuration
- Backing / Managed Beans
- Value & Method Binding
- Actions, Events, Listeners
- Navigation
- Converters & Validators
- Custom components
- Renders

# What Is JSF?

- A Java specification
- API for building Web UI Components
- MVC Based
- Very “Struts Like”
- A Server side, event-driven programming model
- A model to enable 3<sup>rd</sup> party components.

# History

- JSR 127
  - Expert Group formed 5/2001
  - Final Release 3/2004
- Simplify Web Development
- Tool Vendors to provide components for developers to rapidly build web apps, and focus on business logic.



\*test.jsp

test.jsp - test.jsp \*

hx:commandExButton

Standard

Enter something here

Submit

DesignSourcePreview

Quick Edit

Quick Edit [hx:commandExButton] - javascript

Command

onblur

onChange

onclick

ondblclick

```
//use 'thisObj' to refer directly to this component in
//use 'thisEvent' to refer to the event generated inst
```

Properties

ServersConsoleProblems

BODY

hx:scriptCollector

h:form

hx:commandExButton

Display options

Parameter

Accessibility

Id: button1

Style: Props:

Classes: commandExButtc

Type: ☒ Submit ☐ Reset

Click to code the action this button performs. Then, identify the pages the action can go to. The outcome returned by the action determines which rule in the map is followed.

When Outcome Is	Go To Page	If Action Is	
<input type="checkbox"/> Rules on this p...			
success	/test.jsp	Any Action	

Add Rule...

Remove Rule

Edit Rule...

Palette

HTML Tags

JSP Tags

Crystal Reports Faces Com...

Faces Components

Data Table

Panel - Group Box

Panels - Tabbed

Panel - Menu Bar

Command - Button

Command - Hyperlink

Link

Input

Input - Text Area

Rich Text Area

Input - Password

☒ Check Box

☐ Check Box Group

☐ Radio Button Group

Combo Box

List Box - Single Select

List Box - Multiple Select

☐ File Upload

Output

☐ Output - Formatted Text

Image

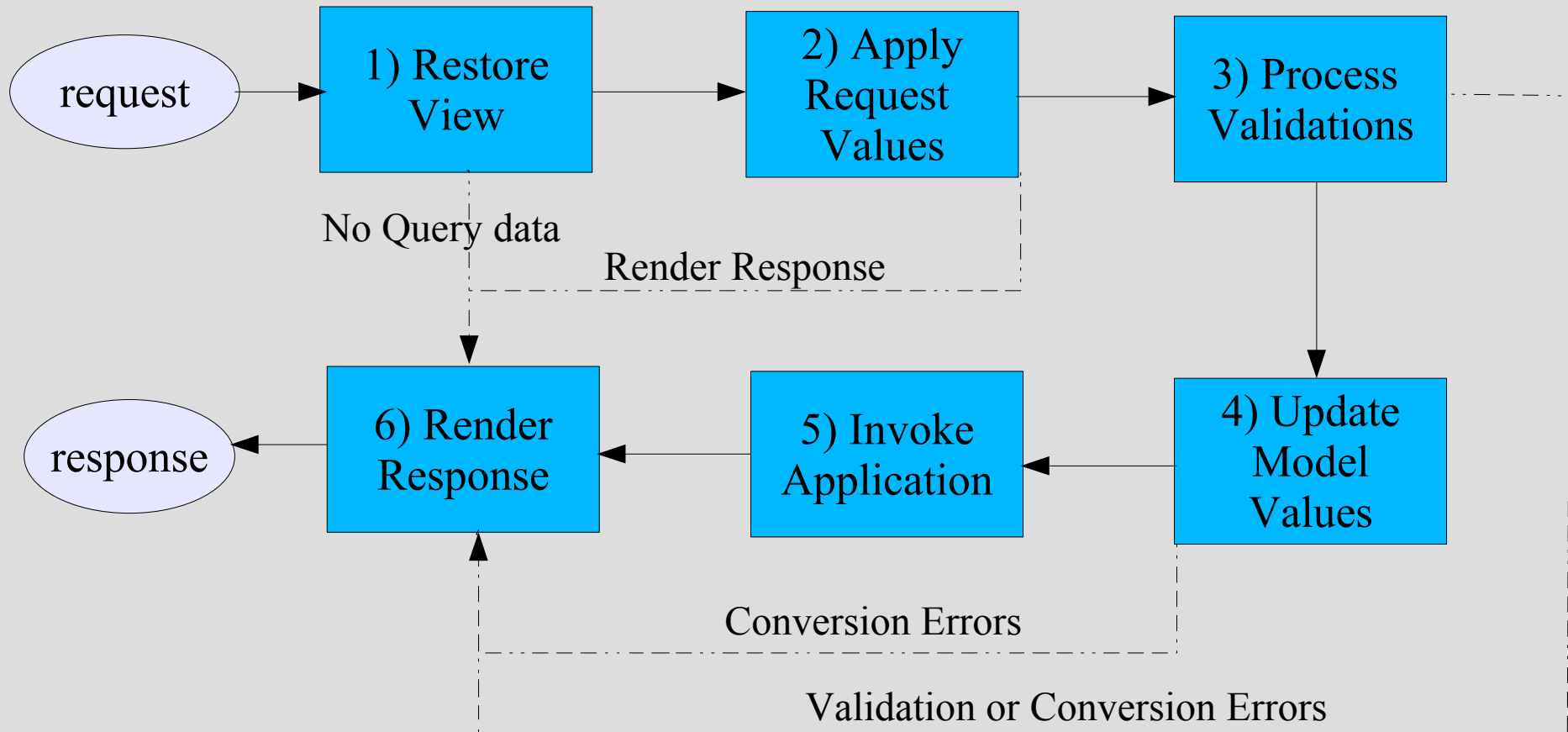
Page Template

Web Site Navigation

Data

# JSF LifeCycle

## 6 Phases



# PhaseListener

- A way to access each phase of the LifeCycle
- Register with the JSF Application.
- Logging example.



# FacesContext / HttpSession

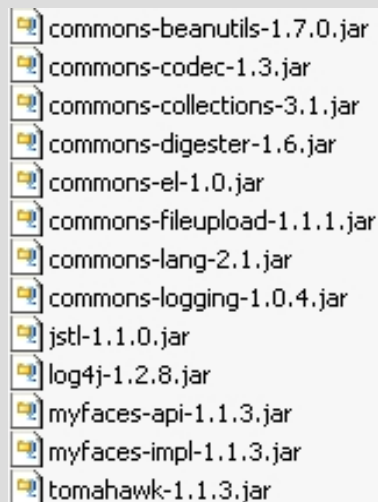
```
FacesContext context =  
    FacesContext.getCurrentInstance();  
  
HttpSession session = (HttpSession)  
    context.getExternalContext().  
    getSession(false);  
  
Map params = context.getExternalContext().  
    getRequestParameterMap();  
  
String action = (String)params.get("foo");
```

# The Demo

- What Demo is Using
  - SDK 1.5
  - Eclipse 3.2
  - Ant 1.6.5
  - JSF 1.1
  - MyFaces 1.1.3
  - Tomahawk 1.1.3 (6/15/2006)
  - Tomcat 5.0.28
  - Servlet 2.4 / JSP 2.0

# Setting Up JSF

- Download myfaces-core-1.1.3-bin.zip
- Configure web.xml
- Configure faces-config.xml



A screenshot of a file explorer window displaying a list of JAR files. The files are listed in a single column, each with a small icon to its left. The files are:

- commons-beanutils-1.7.0.jar
- commons-codec-1.3.jar
- commons-collections-3.1.jar
- commons-digester-1.6.jar
- commons-el-1.0.jar
- commons-fileupload-1.1.1.jar
- commons-lang-2.1.jar
- commons-logging-1.0.4.jar
- jstl-1.1.0.jar
- log4j-1.2.8.jar
- myfaces-api-1.1.3.jar
- myfaces-impl-1.1.3.jar
- tomahawk-1.1.3.jar

# web.xml

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>
    javax.faces.webapp.FacesServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
</servlet-mapping>
```

# faces-config.xml

- All that is JSF!
- Defines
  - Phase-listeners
  - Validators
  - Converters
  - Custom components
  - Managed & backing beans
  - Navigation
  - Others

# Components

- Really just “Fancy Tag” Libraries

- `<f:view>`

...

```
<h:outputLink value="index.jsp">
```

```
  <h:outputText value="Back to  
    Index" />
```

```
</h:outputLink>
```

...

```
</f:view>
```

- Each component tree is denoted by `<f:view>`.  
Many components per tree. One or more trees  
(`<f:view>`) per jsf page.

# Backing / Managed Beans

- Adhere to Java Beans specification
  - Getters / Setters
  - No arg constructor
  - Implement `java.io.Serializable`
- Extends/Implements NOTHING!
- Defined in `faces-config.xml`
- Managed Beans – any bean defined in xml
- Backing Beans – term used to define a bean which “backs” a form

# Value Binding

- Value Binding

- Scriptlet `<%=user.getFirstName() %>`
- UseBean `<jsp:getProperty name=user" property="firstName" />`
- Struts `<bean:write name="user" property="firstName" />`
- JSTL `<c:out value="${user.firstName}" />`
- JSF
  - `<h:outputText value="#{user.firstName}" />`
  - `<h:outputText value="#{expense.en.number}" />`



# ValueBinding (con't)

- JSR 245: JSP 2.1 - Align JSP with JSF
- JSR 252: JSF1.2 - EL
- JSR 45: JSTL 1.2
- Released 5/11/2006, part of Java EE 5
- ```
<c:forEach var="color" items="${colorList}">  
    <h:inputText value="#{color.name}"/>  
</c:forEach>
```

# MethodBinding

- Call a method in a Bean:

```
<h:commandButton value="Save"  
    Action="#{category.addCategoryAction}" />
```

# Demo 1

- Form to create bookmark categories
- Items to Cover
  - Required Tag Libs / components
  - ValueBinding / MethodBinding
  - ActionListener
  - Out-of-box Validation
  - LifeCycle (phase listener)
  - Use of a Backing bean

# Actions

- Similar to Actions in Struts
- Many per page/component tree / many per bean
- Managed Bean didn't extend or implement anything
- Designed for Business Logic
- Determines Navigation

# ActionListeners

Registered to a component

- Provides reusability

```
<h:commandButton value="Cancel"
                  action="#{category.indexAction}"
                  immediate="true" >
    <f:actionListener
        type="com.portal.custom.CancelListener"/>
</h:commandButton>
```

- Also Works (later example):

```
<h:commandButton value="Cancel"
                  action="#{category.indexAction}"
                  immediate="true"
                  actionListener="#{category.someMethod}" />
```

# Navigation

- Specified in faces-config.xml
- Determined by outcome of an action

- Specific to page `<from-view-id >`
- Specific to an Action `<from-action >`
- Supports wild cards

```
<from-view-id>  
    /foo-directory/*  
</from-view-id>
```

- Global

```
<navigation-case>  
    <from-outcome>error</from-outcome>  
    <to-view-id>/error.jsp</to-view-id>  
</navigation-case>
```

# Demo 2

- Form to create Bookmark List
- Items to cover:
  - Use of a Managed Bean
  - Different ways to populate lists
  - ActionListener
  - A little EL
  - ValueChangeListener

# Events

- Like listeners, they adhere to the event model defined in JavaBeans Spec
- 3 kinds
  - **ActionEvent**  
`public void addAnotherBookmarkListener(ActionEvent ae)`
  - **ValueChangeEvent**  
`public void removeItemListener(ValueChangeEvent vce)`
  - **PhaseEvent**  
`public void afterPhase(PhaseEvent arg0)`



# Demo 3

- Form To Enter Expenses
- Items to cover
  - Use of 3<sup>rd</sup> party component
  - Out of box validation / conversion
  - Custom validation / conversion

# Converters

- Convert from String to Object
- Out of Box
  - ConvertDateTime
  - ConvertNumber
- Custom – defined in faces-config.xml

```
<converter>
  <converter-for-class>
    com.portal.expense.EmployeeNumberBean
  </converter-for-class>
  <converter-class>
    com.portal.custom.EmployeeNumberConverter
  </converter-class>
</converter>
```

# Validation

- Defined in faces-config.xml

```
<validator>
  <validator-id>enValidator</validator-id>
  <validator-class>
    com.portal.custom.EmployeeNumberValidator
  </validator-class>
</validator>
```

- In JSP

```
<h:inputText id="en" value="#{expense.en}"
  required="true" >
  <f:validator validatorId="enValidator" />
</h:inputText>
```

# Demo 4

- Simplest Custom Component Ever!
- Items To Cover
  - Setup of Component
  - The JSP
  - The Tag
  - The Component

# Renderers

- Component can delegate another class for rendering the UI
- Use same component across different viewers
- Ex Component w/3 different renderers
  - AJAX Enabled
  - Java Script Disabled
  - WAP

# For More Info

- Core Java Server Faces by David Geary  
Online at <http://corejsf.com>
- <http://jsftutorials.net/>
- <http://myfaces.apache.org/>
- JSF for Non-believers (Rick Hightower)
- [http://www-128.ibm.com/developerworks/views/java/libraryview.jsp?search\\_by=nonbelievers:](http://www-128.ibm.com/developerworks/views/java/libraryview.jsp?search_by=nonbelievers)

# So I don't forget ...

- [mjdecap@yahoo.com](mailto:mjdecap@yahoo.com) - LinkedIn member

- Next WebSphere User Group Meeting:  
Wednesday, August 9, 2006, 4:00 – 5:30pm

Where: OneAmerica Main Auditorium

Topic: Geronimo!

<http://www.WebSphere.org>