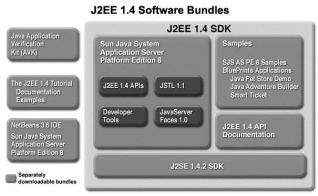
JSF – JavaServer Faces

By Sosuke Tokunaga
CS491b Fall 2004

Introduction

- New Java Technology for Java Server Application
- Bundled with J2EE 1.4 SDK



First Look

- JSF = Struts + Swing

= Java server application framework based on MVC archtecture

+

UI component model for Java server application

Develop server applications in a similar manner with desktop applications

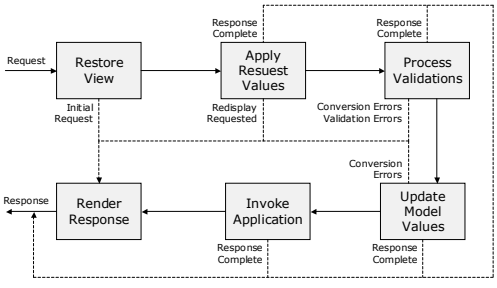
Basic Components

- Faces Servlet
- Faces Context
- UI Components
- Events and Listeners
- Faces Core/HTML Tag Libraries

HTML UI Components

- HtmlCommandButton
- HtmlCommandLink
- HtmlForm
- HtmlInputText
- HtmlInputSecret
- HtmlInputHidden
- HtmlInputTextArea
- HtmlSelectBooleanCheckbox
- HtmlSelectManyCheckbox
- HtmlSelectOneMenu
- HtmlSelectManyMenu
- HtmlSelectOneList
- HtmlSelectManyList
- HtmlSelectOneRadio
- HtmlOutputLabel
- HtmlOutputText
- HtmlOutputLink
- HtmlOutputFormat
- HtmlDataTable
- HtmlGraphicImage
- HtmlMessage
- HtmlMessages
- HtmlPanelGrid
- HtmlPanelGroup
- UIViewRoot
- UIColumn

Faces Request Life Cycle



Phase1: Restore View

- View
  - A UI component tree which represents structured UI components associated with a page
- Initial Request
  - Create a empty view and skip to the Render Response Phase
- Postback
  - Restore the view by using the state information

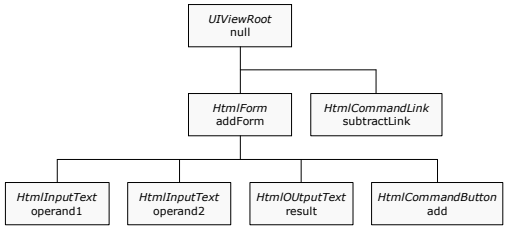
Phase1: Restore View – example(1)

- JSP page with JSF tags

```
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<html>
<head><title>Add 2 Numbers</title></head>
<body>
  <f:view>
    <h:form id="addForm">
      <h:inputText id="operand1" value="#{Addition.operand1}"/> +
      <h:inputText id="operand2" value="#{Addition.operand2}"/> =
      <h:outputText id="result" value="#{Addition.result}"/><br/>
      <h:commandButton id="add" value="Calculate" action="add"
                       actionListener="#{Addition.add}"/>
    </h:addForm>
    <h:commandLink id="subtractLink" action="substact">Go to Subtract</h:commandLink>
  </f:view>
</body>
```

Phase1: Restore View – example(2)

- View – Component Tree



Phase2: Apply Request Values

- For Each Component in the View
  - Retrieve parameters associated with the component
  - Convert the parameters into appropriate type
  - Set the converted value into the component as submitted values
- Redisplay
  - If a event associated with this phase, the handler can skip to the Render Response Phase

Phase3: Process Validation

- Each Component in the View
  - Validates submitted values
  - Sets the submitted values to local values
  - If values have been changed, ValueChangeEvent will be broadcast.
  - Clears the submitted values
- If Errors Occurred
  - Stores error messages into the Context and skips to the Render Response Phase

Phase4: Update Model Values

- Each Component in the View
  - Sets the local values into the referenced properties of Backing Beans
  - Clears the local values
- If Errors Occurred
  - Stores error messages into the Context and skips to the Render Response Phase

Phase5: Invoke Application

- Broadcast Application-level Events
  - Submitting
  - Linking to another page
- Tasks of Event Handlers
  - Invoke services provided by Business Logic
  - Determine the forwarding page displayed next by using NavigationHandler
- Faces Servlet sets the response view of the forwarding page.

Phase6: Render Response

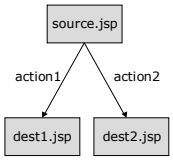
- If Initial Request
  - Add Components on the page into the View
- Rendering
  - Components render themselves onto the page displayed
- Store the state information

Navigation Model

- A Set of Navigation Rules

faces-config.xml

```
<navigation-rule>
   <from-view-id>/source.jsp</from-view-id>
   <navigation-case>
      <from-outcome>action1</from-outcome>
      <to-view-id>/dest1.jsp</to-view-id>
   </navigation-case>
   <navigation-case>
      <from-outcome>action2</from-outcome>
      <to-view-id>/dest2.jsp</to-view-id>
   </navigation-case>
</navigation-rule>
```



Backing Bean Management

- Backing Beans
  - Java Beans whose properties or methods are associated with UIComponents

```
<f:view>
  <h:form id="addForm">
    <h:inputText id="operand1" value="#{Addition.operand1}"/> +
    <h:inputText id="operand2" value="#{Addition.operand2}"/> =
    <h:outputText id="result" value="#{Addition.result}"/><br/>
    <h:commandButton id="add" value="Calculate" action="success"
                     actionListener="#{Addition.add}"/>
  </h:addForm>
  <h:commandLink id="subtract">Go to Subtract</h:commandLink>
</f:view>
```

Backing Bean

Addition.java

```
package cs491.bean

class Addtion {
   private int operand1;
   private int operand2;
   private int result;

   public void Addtion() { operand1 = operand2 = result = 0; }
   public int getOperand1() { return operand1; }
   public int getOperand2() { return operand2; }
   public int getResult() { return result; }
   public void setOperand1(int operand) { operand1 = operand; }
   public void setOperand2(int operand) { operand2 = operand; }
   public void add() { result = operand1 + operand2; }
}
```

Managed Bean Definition

faces-config.xml

```
<managed-bean>
  <managed-bean-name>Addition</managed-bean-name>
  <managed-bean-class>cs491.bean.Addtion</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
```

Installation

- Download Faces package
  - JavaServer Faces v1.1.01 Reference Implementation
    http://java.sun.com/j2ee/javaserverfaces/download.html
  - Unzip the downloaded file
- Copy Faces API and Required API
  - Copy all jar files in $(download)/lib directory into your web application library directory $(WebAppRoot)/WEB-INF/lib

Configration

- 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>

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

Delevopment Steps

- Create JSP page with JSF tags
- Define Page Navigation in faces-config.xml
- Develop the Backing Beans
- Add Managed Bean Declaration in faces-config.xml