

Subject: Web Development with JSF2

Requirement: STS or IntelliJ, Tomcat7, MySql and the JSF “jsf_example”.

Presenter: Ghaith Alrabadi.

Date: 12/10/2012

Web Development with JSF2

What is JSF?

Web Development with JSF2

- JavaServer Faces (JSF) is a Java-based web application framework intended to simplify development integration of web-based user interfaces.
- JSF is a request-driven MVC web framework for constructing user interfaces using components. As a display technology, JSF 2 uses Facelets. JSF 1.x uses JavaServer Pages (JSP) for its display technology.
- JSF is a standardized technology which was formalized in a specification through the Java Community Process. It is part of the Java Platform, Enterprise Edition.
- JSF is event driven programming model.
- JSF is component-driven framework which make it a natural extension of OOD for the developer.

Web Development with JSF2

What is JSF versions?

Web Development with JSF2

Versions:

- JSF 2.1 (2010-10-22) — Current version.
- JSF 2.0 (2009-06-28).
- JSF 1.2 (2006-05-11).

Implementations:

- Mojarra: is the new name of the Oracle original JSF reference implementation.
- Myfaces by apache.

Web Development with JSF2

How JSF works?

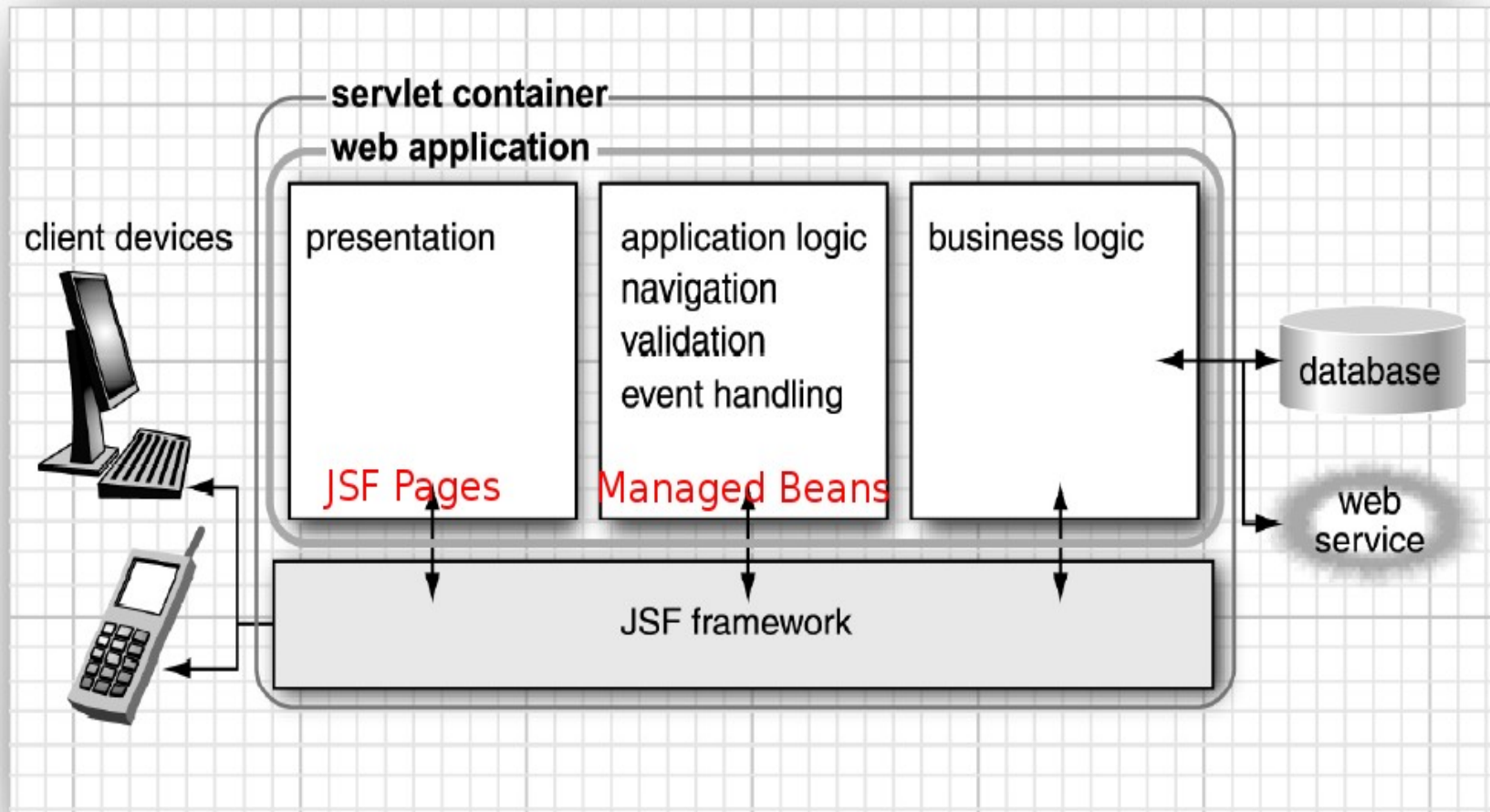
Web Development with JSF2

- JSF is based on a component-driven UI design model, using XML files called view templates or Facelets views.
- Requests are processed by the FacesServlet, which loads the appropriate view template, builds a component tree, processes events, and renders the response (typically HTML) to the client.
- The state of UI components (and some other objects) is saved at the end of each request, and restored upon next creation of that view.
- Several types of state-saving are available, including Client-side and Server-side state saving.

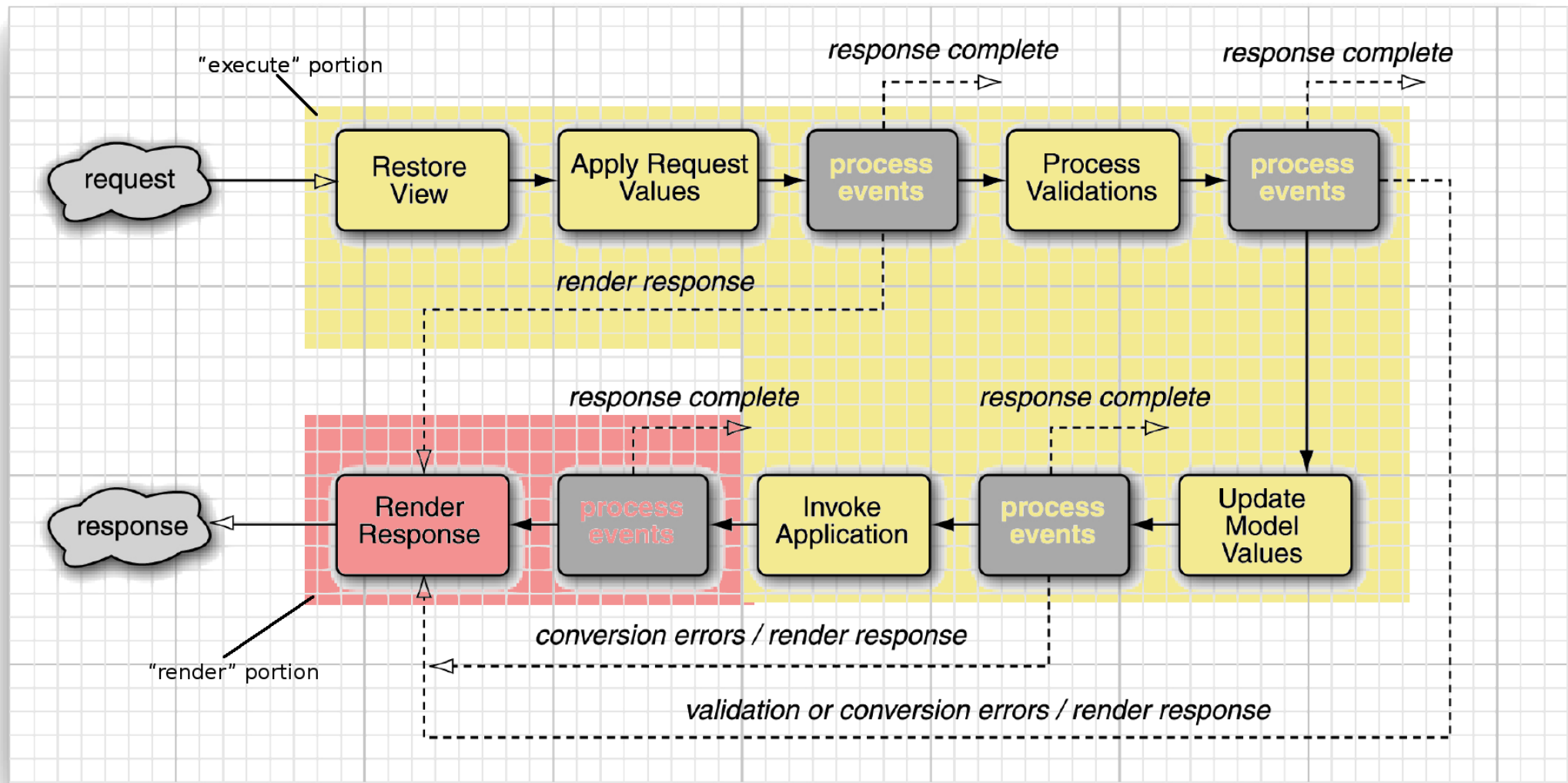
Web Development with JSF2

JSF life Cycle

Web Development with JSF2



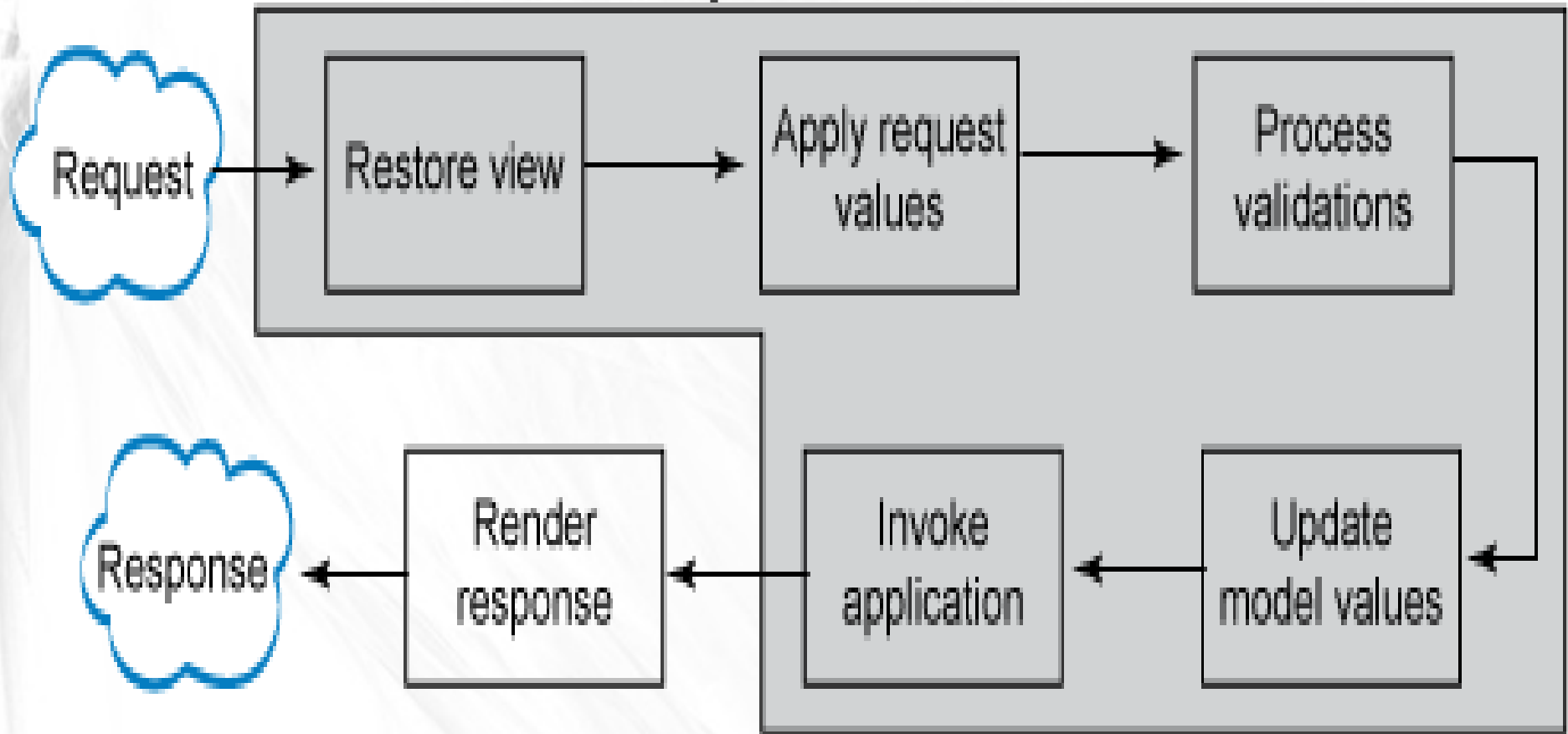
Web Development with JSF2



Web Development with JSF2

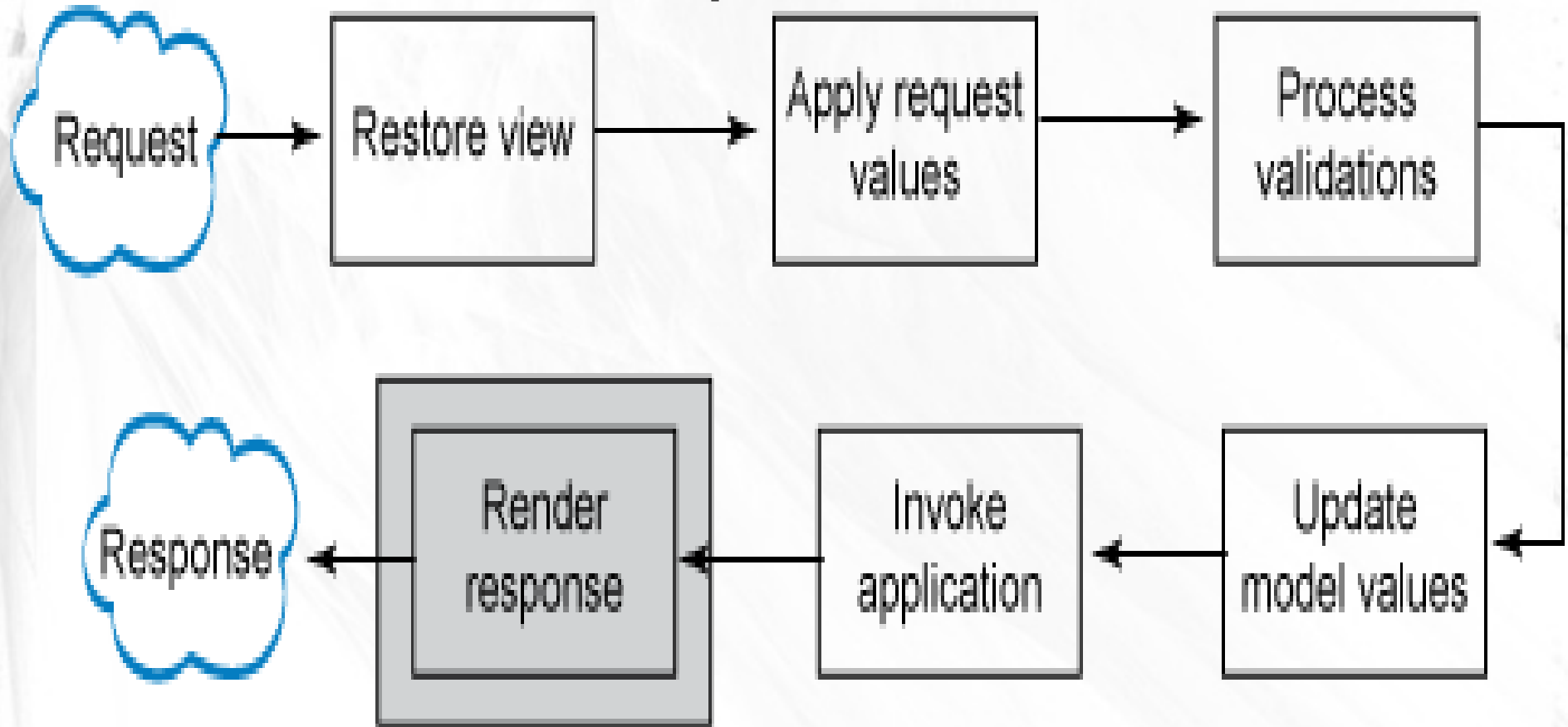
JSF life Cycle for Ajax

Web Development with JSF2



Execute Phase

Web Development with JSF2



Render Phase

Web Development with JSF2

Basic Components and Tags in JSF

Web Development with JSF2

JSF core components are components that are not tied to HTML rendering or any other rendering mechanism. They provide functionality like type conversion and validation, listeners among others.

```
<html xmlns="http://www.w3.org/1999/xhtml"  
      xmlns:h="http://java.sun.com/jsf/html"  
      xmlns:f="http://java.sun.com/jsf/core"  
      xmlns:p="http://primefaces.org/ui">
```


Web Development with JSF2

JSF HTML components are components that are tied to HTML rendering like button, href, select, table ..etc.

```
<html xmlns="http://www.w3.org/1999/xhtml"  
      xmlns:h="http://java.sun.com/jsf/html"  
      xmlns:f="http://java.sun.com/jsf/core"  
      xmlns:p="http://primefaces.org/ui">
```


Web Development with JSF2

JSF Core components Table

Web Development with JSF2

attribute	Sets an attribute (key/value) in its parent component.
param	Adds a parameter child component to its parent component.
facet	Adds a facet to a component.
actionListener	Adds an action listener to a component.
setPropertyActionListener	Adds an action listener that sets a property.
valueChangeListener	Adds a value change listener to a component.
event	Adds a component system event listener.
converter	Adds an arbitrary converter to a component.
convertDateTime	Adds a datetime converter to a component.
validator	Adds a validator to a component.
validateRegex	Validates a value against a regular expression.

Web Development with JSF2

loadBundle	Loads a resource bundle, stores properties as a Map.
selectitems	Specifies items for a select one or select many component.
selectitem	Specifies an item for a select one or select many component.
ajax	Enables Ajax behavior for components.

Web Development with JSF2

JSF HTML components Table

Web Development with JSF2

An Overview of the JSF HTML Tags

- Inputs (input...)
- Outputs (output..., graphicImage)
- Commands (commandButton and commandLink)
- GET Requests (button, link, outputLink)
- Selections (checkbox, listBox, menu, radio)
- HTML pages (head, body, form, outputStylesheet, outputScript)
- Layouts (panelGrid, panelGroup)
- Data table (dataTable and column); see Chapter 6
- Errors and messages (message, messages)

Web Development with JSF2

head	Renders the head of the page
body	Renders the body of the page
form	Renders a HTML form
OutputStylesheet	Adds a stylesheet to the page
outputScript	Adds a script to the page
inputText	Single-line text input control
InputTextarea	Multiline text input control
InputTextarea	Multiline text input control
InputSecret	Password input
inputHidden	Input hidden
outputLabel	Label for another component for accessibility
outputLink	Link to another web site
outputFormat	Like outputText, but formats compound messages
outputText	Single-line text output

Web Development with JSF2

commandButton	Button: submit, reset, or pushbutton
commandLink	Link that acts like a pushbutton
button	Button for issuing a GET request
link	Link for issuing a GET request
message	Displays the most recent message for a component
messages	Displays all messages
graphicImage	Displays an image
selectOneListbox	Single-select listbox
selectOneMenu	Single-select menu
selectOneRadio	Set of radio buttons
selectBooleanCheckbox	Checkbox
selectManyCheckbox	Set of checkboxes
selectManyListbox	Multiselect listbox
selectManyMenu	Multiselect menu

Web Development with JSF2

panelGrid	Tabular layout
panelGroup	Two or more components that are laid out as one
dataTable	A feature-rich table control
column	Column in a dataTable

Web Development with JSF2

General Project Components

Web Development with JSF2

- **JAR files**

- JSF 2.0 JAR files required; JSTL 1.2 JARs some application servers provide them by default like Glassfish 3, JBoss 6, and other Java EE 6 servers

- **faces-config.xml**

mappings of action controller values to results pages. Later tutorial sections will look at explicit values in faces-config.xml.

- **web.xml**

- Must have a url-pattern for *.jsf (or other pattern you choose)
- Usually sets PROJECT_STAGE to Development

- **some-page.xhtml**

- Use URL some-page.jsf (matches url-pattern from web.xml)

Web Development with JSF2

faces-config.xml file:

```
<faces-config
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-facesconfig_2_0.xsd"
  version="2.0">
  <!-- JSF and Spring are integrated -->
  <application>
  <el-resolver>
    org.springframework.web.jsf.el.SpringBeanFacesELResolver
  </el-resolver>
</application>
<!-- configuration of navigation rules -->
<navigation-rule>
  <from-view-id>/pages/index.xhtml</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/pages/success.xhtml</to-view-id>
  </navigation-case>
</navigation-rule>
</faces-config>
```

Web Development with JSF2

web.xml file:

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>JSF_Example</display-name>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
      /WEB-INF/applicationContext.xml
    </param-value>
  </context-param>
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>
```

Web Development with JSF2

```
<welcome-file-list>
  <welcome-file>/pages/index.xhtml</welcome-file>
</welcome-file-list>
<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>*.jsf</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
</servlet-mapping>
</web-app>
```

Web Development with JSF2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html">
<h:head>
    <title>Welcome</title>
</h:head>
<f:metadata>
    <f:viewParam name="directLogin" value="#{user.canLogin}"/>
</f:metadata>
<h:body>
<h:form>
    <h3>Please enter your name and password.</h3>
    <table>
        <tr>
            <td>Name:</td>
            <td><h:inputText value="#{user.name}"/></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><h:inputSecret value="#{user.password}"/></td>
        </tr>
    </table>
    <p><h:commandButton value="login" action="#{user.login}"/></p>
</h:form>
</h:body>
</html>
```

Web Development with JSF2

```
import java.io.Serializable;
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;

@Named("user") // or @ManagedBean(name="user")// or Spring Component i
@SessionScoped
public class UserBean implements Serializable {
    private String name;
    private String password;
    private boolean directLogin;
    public String getName() { return name; }
    public void setName(String newValue) { name = newValue; }
    public String getPassword() { return password; }
    public void setPassword(String newValue) { password = newValue; }
    public boolean isDirectLogin() { return directLogin; }

    public void setDirectLogin(boolean newValue) {
        if(directLogin)
            return "success";
        directLogin = newValue;
    }

    public String login(){
        if(name.equals("a"))
            return "success"
        return "fail"
    }
}
```

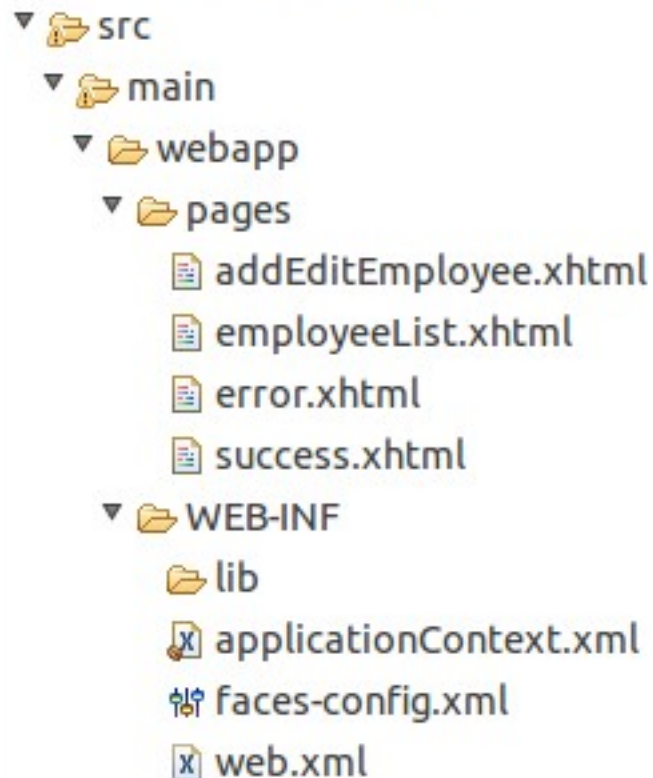
Web Development with JSF2

JSF with Spring and Hibernate Maven Structure

Web Development with JSF2

- ▼ jsf_example
 - ▶ src/main/resources
 - ▼ src/main/java
 - ▼ com.sourceallies
 - ▼ action
 - ▶ EmployeeAction.java
 - ▼ service
 - ▼ dao
 - ▶ EmployeeDAO.java
 - ▶ EmployeeDAOImpl.java
 - ▼ model
 - ▶ Customer.java
 - ▶ Employee.java
 - ▶ Office.java
 - ▶ Order.java
 - ▶ Orderdetail.java
 - ▶ OrderdetailPK.java
 - ▶ Payment.java
 - ▶ PaymentPK.java
 - ▶ Product.java
 - ▶ Productline.java
 - ▶ EmployeeService.java
 - ▶ EmployeeServiceImpl.java

Web Development with JSF2



Web Development with JSF2

Add Tomcat web-server to STS

Web Development with JSF2

Add 'jsf_example' Project to STS

Web Development with JSF2

Create the DB schema on MySql

Web Development with JSF2

Understand the project source code

Web Development with JSF2

Running the Project

Web Development with JSF2

Thank you