# JavaServer Faces: An Overview

Prepared by:
Amanj Mahmud Sherwany

2010

# Agenda 1

- What is JavaServer Faces (JSF)?
- JSF vs neighbor technologies
- IDE Selection
- JSF after Oracle-Sun deal
- JSF Life-cycle
- JSF request/response
- JSF components

# Agenda 2

- MVC and JSF
- Hello World Example
- Pros of JSF
- Cons of JSF
- Conclusion
- References

# What is JSF?

- A set of APIs for representing UI components and managing their state, handling events and input validation, defining page navigation, and supporting internationalization and accessibility.
- It can be experienced within JSP, by JSF custom tag library.

# JSF vs JSP

- JSF is usually used with JSP or Servlets.
- JSP doesn't have UI components.
- Both of them follow XML syntax style.
- JSF unlike JSP has a very complex life-cycle.

# JSF vs GWT

- JSF runs in the server side.
- GWT runs in the client side.
- GWT, is a Java code, which translates to its equivalent JavaScript code.
- JSF, is a script code, which translates to its equivalent Java code.
- Both perform well in supporting AJAX.

# JSF vs AJAX

- AJAX stands for, Asynchronous JavaScript and XML.
- AJAX runs in the client's browser.
- AJAX doesn't need any special program to run on, except a decent browser.
- JSF needs a Servlets container in the server side.
- AJAX unlike JSF is a combination of JavaScript, CSS, DOM and XML.

# JSF vs RIA

- RIA, needs some plugin (or run time) in the client side, to run on.
- JSF doesn't require the client to install any special program, except a decent browser.
- RIA can be written in Adobe Flex, MS Silverlight, JavaFX, … etc!

# IDE Selection 1

When it comes to Java, you have a wide range of options, here is the summary:

- JDeveloper by Oracle: It is one of the best IDEs for JavaEE, the IDE can be freely downloaded, despite its heavyweight it is not open source!
- Netbeans IDE: Again, its a superb IDE for JavaEE, it is a real lightweight IDE (compared to JDeveloper) has a wide range of plugins, and most importantly it is a free open source IDE.

# IDE Selection 2

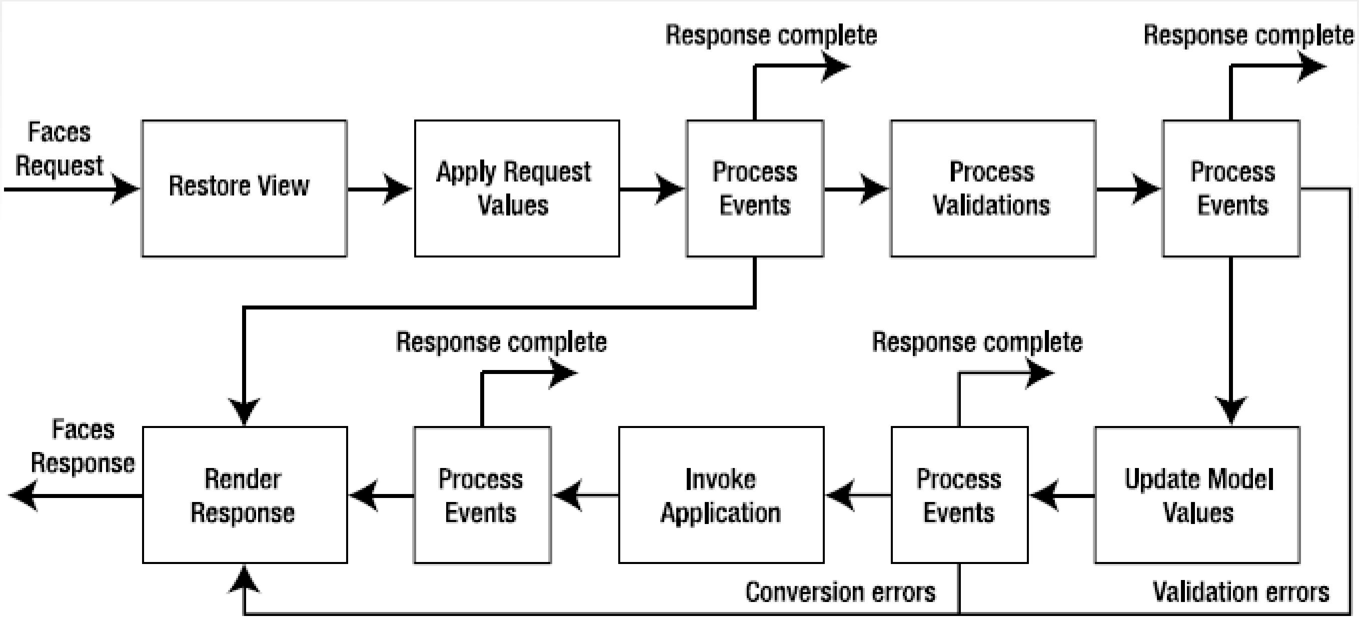
- Eclipse IDE: Another good IDE, it is a free and open source product, it is highly customizable via its wide range of plugins.

- IntelijIDEA: Some say it is the best Java IDE, it has two versions community and ultimate.

# JSF After Oracle-Sun Deal

- JSF is an open source product.
- Oracle was a key player in supporting JSF.
- Oracle promises its customers to spend more money in investing JSF than what Sun did!
- But nothing can guarantee us what is going to happen in the post Oracle era!

# JSF Life-cycle



# JSF Request/Response

- Non-JSF request generates JSF response.
- JSF request generates JSF response.
- JSF request generates non-JSF response.

# JSF Components

- JSF has two type of actions (elements):
  - HTML Custom Actions: includes Input, Output, Selection, Commands & Miscellaneous.
  - Core Custom Actions: includes Converters, Listeners, Miscellaneous, Selection, Validators, View.

# HTML Custom Actions

- Input: <h:inputHidden/>, <h:inputSecret/>, <h:inputText/>, ...etc
- Output: <h:message/>, <h:outptuFormat/>, <h:outputLink/>, ...etc
- Selection: <h:selectBooleanCheckbox/>, <h:selectOneRadio/>, ...etc
- Commands: <h:commandButton/>, <h:commandLink/>, ...etc
- Miscellaneous: <h:dataTable/>, <h:form/>, ...etc

# Core Custom Actions

- Converters: <f:convertDateTime/>, <f:convertNumber/>, ...etc
- Listeners: <f:actionListener/>, <f:valueChangeListener/>
- Miscellaneous: <f:loadBundle/>, ...etc
- Selection: <f:selectItem/>, ...etc
- Validators: <f:validateLongRange/>, ...etc
- View: <f:view/>, ...etc

# MVC and JSF

- JSF supports MVC architecture by inheritance.
- In JSF, the MVC architecture is:
  - Model (the M part): Managed Beans (aka, JavaBean classes).
  - View (the V part): JSF enabled pages
  - Control (the C part): XML files
- You can't program in JSF without taking an MVC approach!

# Hello World Example 1

index.jsp (to generate the JSF response):

```
<%@page pageEncoding="UTF-8"%>

<jsp:forward page="welcomeJSF.faces"/>
```

# Hello World Example 2

The welcomeJSF.jsp (to process the JSF response):

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>

<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
   Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<f:view>

  <html>

    <head>
```

# Hello World Example 3

```
            <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
            <title>JSF by Example</title>
        </head>
        <body>
            <h1><h:outputText value="Hello World"/></h1>
        </body>
    </html>
</f:view>
```

# Hello World Example 4

web.xml (to direct everything with faces as an extension to servlets Container):

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5" 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-app_2_5.xsd">

   <servlet>

      <servlet-name>Faces Servlet</servlet-name>

      <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
```

# Hello World Example 5

```
<load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
      <servlet-name>Faces Servlet</servlet-name>
      <url-pattern>*.faces</url-pattern>
  </servlet-mapping>
  <session-config>
      <session-timeout>30</session-timeout>
  </session-config>
```

# Hello World Example 6

```
<welcome-file-list>
    <welcome-file>forwardToJSF.jsp</welcome-file>
   </welcome-file-list>
</web-app>
```

# Hello World Example 7

faces-config.xml (to configure the jsf files behavior):

```
<?xml version='1.0' encoding='UTF-8'?>
<faces-config version="1.2"
    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_1_2.xsd">
</faces-config>
```

# Hello World Example 8

- All that for one simple application?
- Well no, you don't need to do anything manually except the thing in the welcomeJSF.jsp file.
- A smart IDE can do the whole nasty configurations for you.
- Lets see the Demo first.

# Pros of JSF 1

- Enables non-Java programmers to program in it.
- Has a strong IDE support.
- Has a good support for AJAX.
- Eases web programming.
- Like any other Java based technologies, it has a strong support for Internationalization and Localization.

# Pros of JSF 2

- It supports multi-threading by nature, therefore, there is no need to worry about threadsafe programming.
- Is an open source technology.
- Has a strong support from big companies.
- Doesn't depend on client's web browser.

# Cons of JSF

- It is almost impossible to program without having a smart IDE.
- Needs a Servlets container in the server side.
- Every action is done in the server side (well, this can be pro as well).
- The JSF standard doesn't support GET (but is this really a minus?).

# Conclusion

- JSF differs from JSP and other client side technologies, like GWT, AJAX and RIA.
- JSF can't replace client side technologies.
- Using JSF makes programming a lot easier.
- JSF can be combined with AJAX and other client side technologies.
- JSF is suitable for both Java and non-Java programmers.
- JSF is tool friendly.

# References

- http://java.sun.com/javaee/javaserverfaces/
- http://java.sun.com/javaee/javaserverfaces/reference/docs/
- Beginning JavaEE 5 Platform, K. Mukhar & C. Zelenak

# Questions?