

# JBoss Community



**RichFaces**

**Prabhat Jha**

Principal Engineer  
JBoss, by Red Hat

# Game Plan

- Richfaces History
- A Quick Recap of JSF
- Core Richfaces Concepts & Features
- Overview of the Component Library
- Moving to JSF2

# History

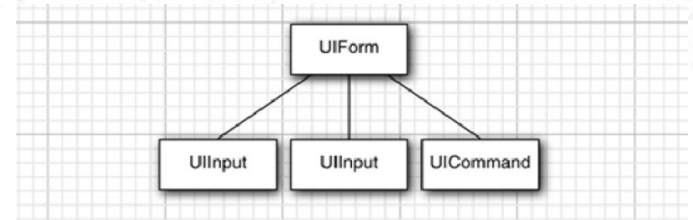
- 2005-2006: ajax4jsf framework introduced
- 2006/2007: ajax4jsf becomes OSS & part of dev.java.net, Exadel built Richfaces components library on top of the framework
- 2007 – Partnership between Red Hat & Exadel
  - A4j & rich merged to single project
  - Richfaces open sourced
- 2009 – Richfaces became a JBoss product.

# RF summary in one slide

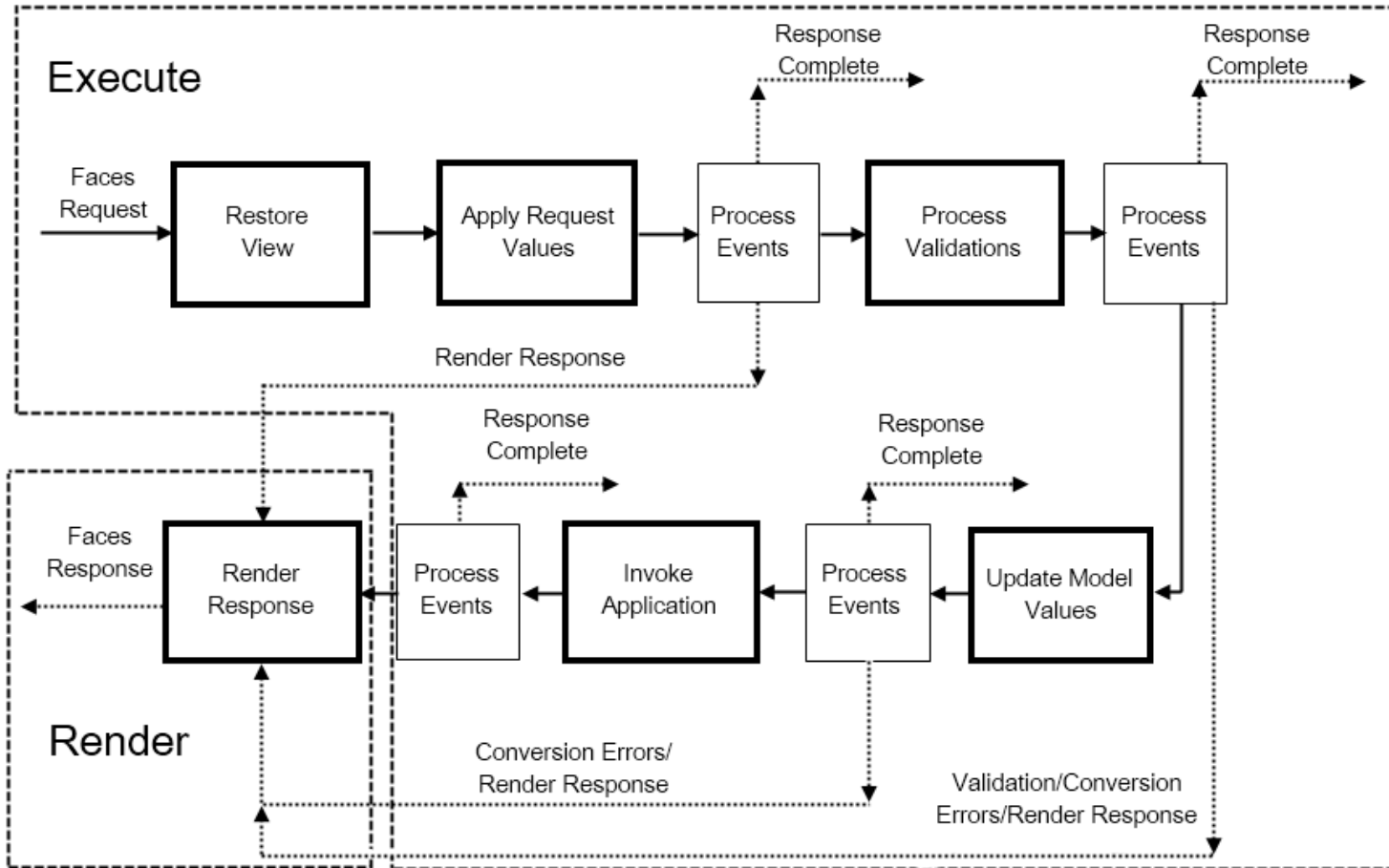
- RichFaces is a JSF components library NOT an implementation of JSF like Mojarra & Myfaces.
- Adds Ajax support to JSF components
- Provides OOB set of rich JSF components
- Skins & Themes
- Dynamic Resource Handling
- Component Development Kit (CDK)
- That's it. Let's go home. :-)

# JSF Intro in 2 slides..(1/2)

- Components
  - Basic UI building block in JSF
  - Assembled into component tree (view)
- Validators
  - Check correctness of user inputs
- Renderers
  - Output components to the client
- Converters
  - Responsible for data conversion between strings and another data types (in both ways)
- Backing beans
- Listeners
- Events



# JSF Intro in 2 slides..(2/2)



# Richfaces Internals

Let's go for a deeper dive

# Tag Libraries

- Two tag libraries
  - **a4j** – provides page wide Ajax functionality as well as other utility (poll, push, jsFunction) components.
  - **rich** – rich UI components
    - Built-in Ajax
    - Highly customizable



# a4j:

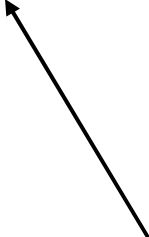
## Need to define:

- When to update (*mouse, key, selection*)
- What to update (*business method*)
- Where to update (*elements on page*)

## a4j:support – attach Ajax request to any JSF component

```
<h:selectOneMenu value="#{bean.fruit}">
  <a4j:support event="onchange"
    action="#{bean.change}"
    reRender="info, details"/>
</h:selectOneMenu>
<h:panelGrid id="info">
  ...
</h:panelGrid>
<h:panelGrid id="details">
  ...
</h:panelGrid>
```

Can also be an  
EL-expression:  
reRender="{bean.areas}"



# Won't you flood server though?

- Tips:
  - eventsQueue
  - requestDelay
  - IgnoreDupResponses
- ajaxSingle=true (single component processing)
- similarityGroupingId

## a4j:jsFunction – Ajax request from any JS function

```
<table>
  ...
  <td onmouseover="update( 'yellow' )"/>
  ...
</table>
<a4j:jsFunction name="update"
                action="#{bean.change}"
                reRender="panel">
  <a4j:actionparam value="param1"
                  assignTo="#{bean.color}"/>
</a4j:jsFunction>
```

## a4j Container Components

- Containers
  - **region** – processing limitations (server)
    - default is whole page
    - can be nested
  - **outputPanel** – update zones definitions (client)
  - **include** – Ajax navigation within subview
    - for wizard like behavior

*Note: Don't get confused: region defines which component will be processed. Page update would still have to be done with reRender or outputPanel.*

## a4j:outputPanel – auto-rendered area

```
<h:selectOneMenu value="#{bean.fruit}">
  <a4j:support event="onchange"
    action="#{bean.change}"/>
</h:selectOneMenu>
<a4j:outputPanel ajaxRendered="true">
  <h:panelGrid>
    ...
  </h:panelGrid>
  <h:panelGrid>
    ...
  </h:panelGrid>
</a4j:outputPanel>
```

- Updates on any Ajax request
- Add/Remove operations in DOM
- Storing transient elements in JSF tree

## Pop Quiz

How are a4j:region component & ajaxSingle attribute related?

# a4j:poll & a4j:push

## Poll:

```
<h:form>
```

```
  <a4j:poll id="poll" interval="1000" enabled="#{userBean.pollEnabled}"
    reRender="poll,grid"/>
```

```
  .....
```

```
</h:form>
```

## Push:

```
<h:form>
```

```
  <a4j:push reRender="msg" eventProducer="#{pushBean.addListener}" interval="2000"/>
```

```
<a4j:outputPanel id="msg">
```

```
  <h:outputText value="#{pushBean.date}"/>
```

```
</a4j:outputPanel>
```

```
</h:form>
```



# a4j:queue

```
<h:form>  
  <a4j:queue size="2" requestDelay="500" sizeExceededBehavior="dropNext"  
  onsizeexceeded="alert('The size of the queue is exceeded')" />  
  <h:inputText value="#{bean.a}">  
    <a4j:support event="onkeyup" />  
  </h:inputText>  
  .....  
</h:selectBooleanCheckbox>  
</h:form>
```

## Scopes:

- Application-wide queue
- View and form based (only one active at a time queues)
- Named queues for request groups

# a4j:queue

Other queue size exceeded behaviors:

- "dropNext" drops next request that should be fired
- "dropNew" drops the incoming request
- "fireNext" immediately fires the next request in line to be fired
- "fireNew" immediately fires the incoming request.

# a4j: miscellaneous componets

- Resource handling
  - `loadStyle` and `loadScript`
  - `mediaOutput`
- Indication
  - `status`
- Client side debug
  - `log`
- Ajax Events
  - `ajaxListener`

# rich: Validators

- Ajax Validator
  - Skips phases 4 & 5
- Bean Validator
  - Model based constraints defined using Hibernate Validator
- Graph Validator
  - Validate all components in a component subtree

# rich: beanValidator

```
<h:inputText value="#{bean.email}">  
  <rich:beanValidator/>  
</h:inputText>  
<rich:message for="email"/>
```

```
public class Bean {  
  @Email  
  private String email;  
  // setter and getter  
}
```

# Rich Skins

- Application-wide skinning support
- Advanced control using predefined skins
- Support for standard and 3<sup>rd</sup> party skinning components
- Archetype

Expenses		
Meals	Hotels	
\$37.74	\$112.00	\$45.00
\$27.28	\$112.00	\$45.00
\$65.02	\$224.00	\$90.00

# rich: Rich components

- Data Iteration Components
- Drag Drop
- Rich Menu
- Rich Trees
- Rich Output
- Rich Input
- Rich Selects
- Rich Validators

# Data Iteration Components

## 3 Major table components

- **DataTable**
  - Row & Column Spanning
  - Flexible layout for header and footer
- **ScrollableDataTable**
- **ExtendedDataTable**
  - Ajax rows loading on scroll
  - Resize
  - Grouping by Column
  - Selection

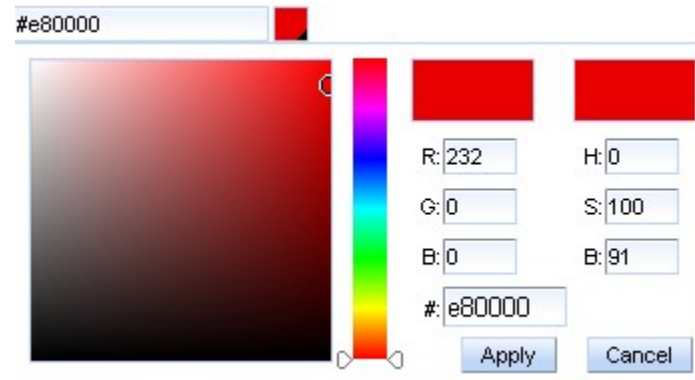
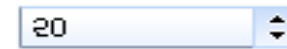
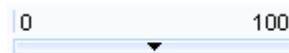
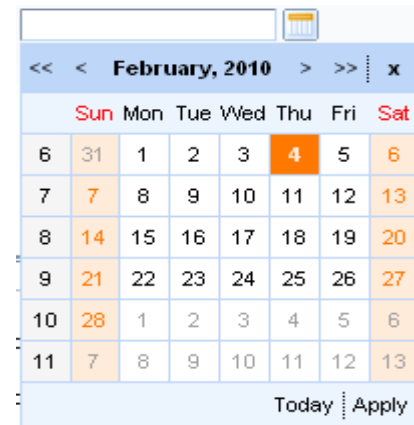
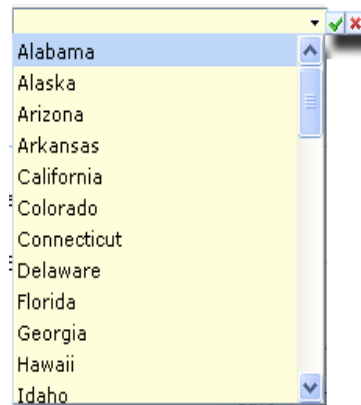
## Common Features

- Filtering
- Sorting
- Pagination
- Cells Ajax Updates
- Events for rows



# Rich Inputs

Inplace Input  
Inplace Select  
Input Number Slider  
Input Number Spinner  
Input Number Spinner  
Suggestion Box  
Calendar  
File Upload  
Color Picker



# Rich Containers & Menu

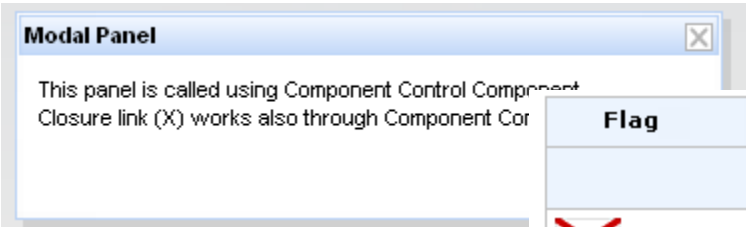
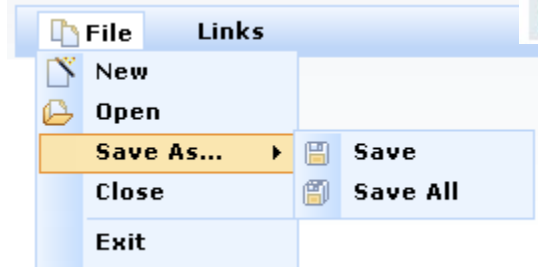
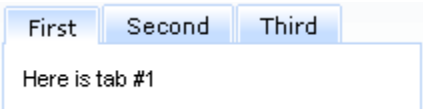
Write your own custom rich components with built-in AJAX support

Package resources with the application's Java classes

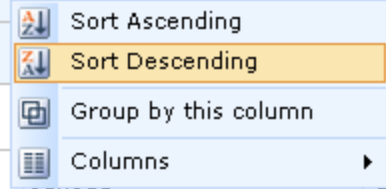
Easily generate images on-the-fly

Resource framework can generate images on-the-fly so that it becomes possible to create images using the familiar approach of the Java graphic2D library.

Create a modern rich user interface look-and-feel with skins-based technology



Flag	State Name ↓	State Capital ▲	Time Zone
	Alabama		MT-6
	Alaska		MT-9
	Arizona	Phoenix	GMT-7

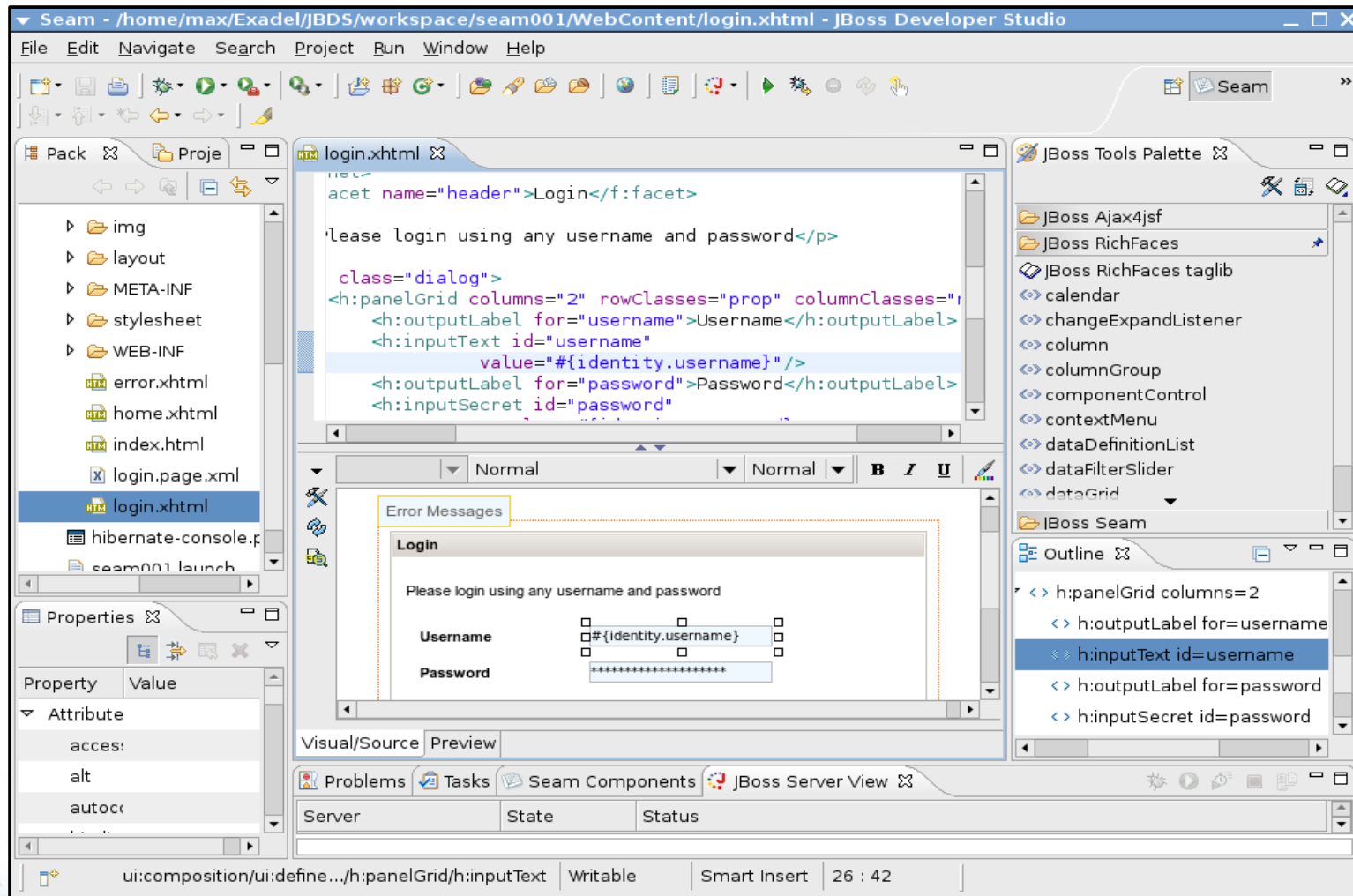


# Component Dev Kit (CDK)

## Tasks for JSF component developer

- Component class implementation
  - Components attributes handling
  - Save-restore component state
- HTML prototype
- Renderer implementation
  - HTML code rendering
- Client-side code development
- Support for converters/listeners
- Creation of Tag, TagHandler classes
- Creation of \*.tld for JSP and \*.taglib.xml for Facelets
- Unit tests development

# JBoss Tools/JBoss Dev Studio



# Richfaces config

## web.xml

```
<filter>
  <display-name>RichFaces Filter</display-name>
  <filter-name>richfaces</filter-name>
  <filter-class>org.ajax4jsf.Filter</filter-class>
</filter>
<filter-mapping>
  <filter-name>richfaces</filter-name>
  <servlet-name>Faces Servlet</servlet-name>
  <dispatcher>REQUEST</dispatcher>
  <dispatcher>FORWARD</dispatcher>
  <dispatcher>INCLUDE</dispatcher>
</filter-mapping>
```

## WEB-INF/lib

```
richfaces-api-3.3.X.jar
richfaces-impl-3.3.X.jar
richfaces-ui-3.3.X.jar
```

OR  
Use maven  
archetypes

## .xhtml:

```
xmlns:rich="http://richfaces.org/rich"
xmlns:a4j="http://richfaces.org/a4j"
```

# Troubleshooting

- Main principle of solving problems in RichFaces based applications :“No magic” ☺
- Remember it’s still just JSF components and just standard JSF lifecycle for Ajax requests
- Server side
  - Problems with architecture in general or with application code
  - Server-side logging
  - Faces Messages
  - Just debug your code!
- Client Side
  - a4j:log
  - FF FireBug, IE Dev toolbar, Fiddler
- Finally if nothing helps – <http://community.jboss.org> we will be happy to help

# JSF 1.2 Pros & Cons

## Pros

- Standards based
- Well defined request lifecycle
- Designed with tooling in mind
- Component-oriented approach
  - Custom UI widgets
- Many extension points
  - Component libraries
  - Phase listeners
  - View handlers
  - Pluggable renderers
  - ...

## Cons

- JSP for View Layer
- No built in AJAX support
- XML hell
- Custom Component complex to create
- Not easy to bookmark URL

# JSF 2 : We can sleep better

- View Declaration Language (VDL) & Facelets
- Composite components
- Behaviors
- Built-in AJAX
- Navigation enhancements
- New bean scopes
- Configuration enhancements
- Project stages
- Validation enhancements
- Exception handling
- Resources loading
- System events



# JSF 2 : JBoss Contribution

- Ajax support (f:ajax)
- Bean Validation Integration
- View parameters
- Bookmarkable links and buttons
- <f:metadata> component
- Exception handling
- Behaviors
- ....

# Ajax in Richfaces 3 & JSF2

```
<h:inputText value="#{bean.text}" >  
    <a4j:support event="onkeyup"  
        action="#{bean.countAction}"  
        reRender="id1, id2" />  
</h:inputText>
```

RichFaces 3.x

```
<h:inputText value="#{bean.text}" >  
    <f:ajax event="keyup" execute="@form"  
        listener="#{bean.countListener}"  
        render="id1 id2" />  
</h:inputText>
```

JSF 2

# Richfaces 4 & JSF2

- Currently in Alpha Stage
- Core & CDK are mostly complete
- A set of components are already migrated and more are being added in every release
- Will tell you more when time is right. :-)

# Follow & Contribute

- Project Page: [www.jboss.org/richfaces](http://www.jboss.org/richfaces)
- @richfaces
- IRC on Freenode: #richfaces
- Live Demo: <http://exadel.com/web/portal/richfacesdemo>
- Source: <http://anonsvn.jboss.org/repos/richfaces/>
- User forum: <http://community.jboss.org/en/richfaces?view=discussions>

# Personal Appeal



<http://www.eejot.org>