



J-Fall

11 oktober 2007 Spant! - Bussum



Real world usage of AJAX and JSF

Tiese Barrell
Adriaan Wisse



Agenda

- JSF and AJAX recap
- User experience
- Why use AJAX frameworks?
- Frameworks positioning
- Demonstration of framework usage
- Frameworks pros and cons
- Questions



Recap: JSF – Java Server Faces

- API for components creation & usage and TLD for a set of components
- Separation of concerns:
 - Component state (M)
 - Component presentation (V)
 - Component behavior (C)
- Lets the **application developer** focus on creating the **application** and the **web developer** focus on creating **pages**

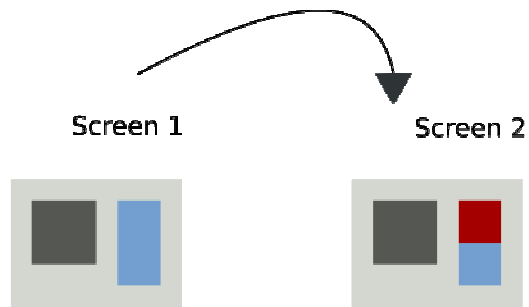


Recap: AJAX

- Asynchronous JavaScript and XML
- Leverages **existing** technologies to provide more application-like responsiveness and interactivity to web applications

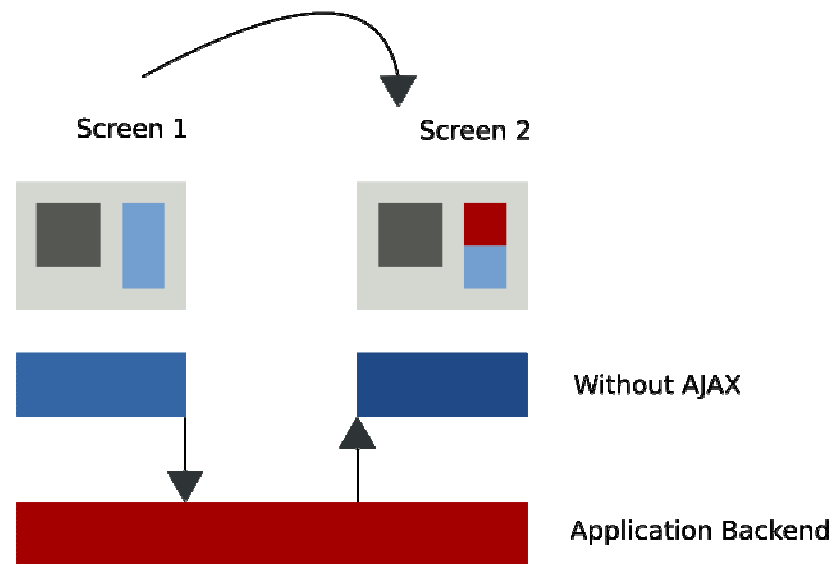


Recap: AJAX



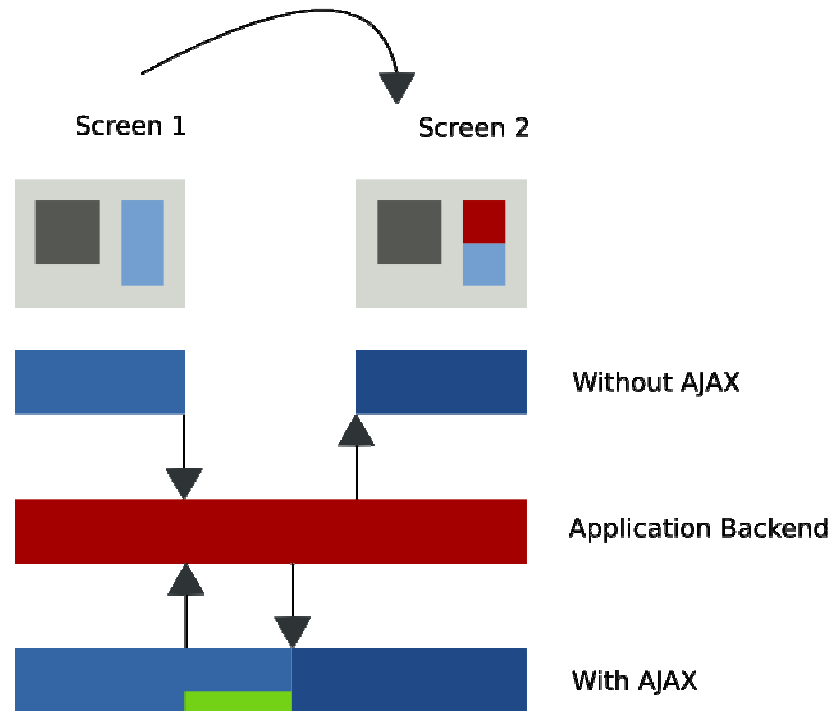


Recap: AJAX





Recap: AJAX





Recap: JSF & AJAX

- Do JSF and AJAX play nicely?
- Standard JSF model is based on form submit and full page rendering, even if only part of the UI may need to change
- For AJAX we require Partial Page Rendering (PPR) and XML exchange
- Two approaches:
 - AJAX enabled components
 - Adding AJAX functionality to existing components



Agenda

- JSF and AJAX recap
- **User experience**
- Why use AJAX frameworks?
- Frameworks positioning
- Demonstration of framework usage
- Frameworks pros and cons
- Questions



User experience (UE)

- User experience encompasses all aspects of the end-user's interaction with the company, its services, and its products. *
 - Meet the exact needs of the customer
 - Simplicity and elegance: a joy to own, a joy to use
- ... pertains to the creation of the architecture and interaction models which impact a user's perception of a device or system. **
 - User's interaction with the product: how it is perceived, learned, and used.
- Experience matters: the fact is, in today's world, there are many products out there that can do what we need but few that allow us to do it quickly, elegantly and memorably. ***
 - Engaging, interactive, lightweight, and flexible.

* Nielsen Norman Group ** Wikipedia *** Al Ramadan, Macromedia



UE paraphrased

64	Rate of Return Seminar or Course
65	Industrial Project Start-Up, Expansion and Management (1) Seminar or Course
66	Industrial Project Start-Up, Expansion and Management (2) Seminar or Course
66	Anti-Dumping In Practice: Hypothesising, Case Reporting, Case Investigation, Countervailing A Sunset Review and Suspension Seminar or Course

Human Resource and Organisational Development Consultancy (HRODC) Ltd

Human Resource & Organisational Development Consultancy (HRODC) Ltd. - An International Human Resource (HR) Training Provider - is a Human Resource, Business Management and Organisational (Organizational) Development (OD) Consultancy, based in London and Wolverhampton, West Midlands UK, Delivering Human Resource, Research Training, Social Care Solutions, Human Resource (HR) Solutions, HR Outsourcing Facilitation, Social care, Human Resource Management Training Courses, Employee development, Change management, Diversity management, Diversity Management Training, Research Coaching, Research Project Management, Research Consultancy, Research Support, Research Supervision, Business analysis, System Analysis and Design, Web Design, Organisational (Organizational) Development (OD); Systems and Process Design and Implementation; Project Management and related services, Team development - establishment of Autonomous Work Teams or High-performing Teams; Reward Schemes; Workforce Flexibility; Organisational (Organizational) Design - Structuring Organisations, Restructuring organisations (organizations); Problem Resolution; Conflict Management, Work-Life Balance, incorporating Organisational (Organizational) Flexibility; Flexible Working Practices; and High-Performance Teams. Seminar locations include Munich Germany; Durban South Africa, Johannesburg South Africa, Malta, Hong Kong, Muscat Oman, Kuwait City Kuwait, Brunei Darussalam, Kuala Lumpur Malaysia, Lagos Nigeria, Abuja Nigeria, Manila Philippines, Kathmandu Nepal, Mumbai India, Damascus Syria, Port Louis Mauritius, Jeddah, Saudi Arabia, Riyadh Saudi Arabia, Dubai UAE, United Arab Emirates, Paris France, Rome Italy, Athens Greece, Albania, Kazakhstan, Tajikistan, St. Petersburg Russia, Toronto Canada, New York USA, Caracas Venezuela

Scroll Down Page!!



arrangement with th

E-MAIL

HRODC Ltd.

or of Technological U
opening up key aspects to

Other HRODC Site

Click Computers

Card Machines

Card Printing

Web Design

Logo Design

Personnel And Occupations

Antigua, Argentina, Armenia, Aruba
International training seminars or pub
Organisational Change Management
Testing Seminar or Course, Personne
Design and Results Analysis Semina
Seminar, Trainer Training; Training fo
Intrinsic and Extrinsic Rewards Sem
and Selection Seminar, Diversity
Seminar, Research Project: Design, C
and Anti-Subsidy; Claims, Measures
Seminar, Human Resource Managem
Financial Risk Management Seminar,
and financial crimes Seminar, Judge S
Organisation, Management & Sente
Automotive Industry Seminar, Adv
Productivity Improvement Seminar, C
Seminar, Real Estate Management: An
Property Valuation Training Semin
Comprehensive View Seminar, Risk &
Fundamentals of Aviation Cargo Semi
goods by Sea. This web site hosts se
research articles of interest to:
University academics; College ac
researchers; Senior managers; Middle
practitioners; Management consultan



UE paraphrased



Google Search | I'm Feeling Lucky

[Advanced Search](#)
[Preferences](#)
[Language Tools](#)

[Advertising Programs](#) - [Business Solutions](#) - [About Google](#) - [Go to Google Nederland](#)

©2007 Google



UE paraphrased

- AJAX can enhance the user experience
- Applying AJAX is **not** so much about
 - XMLHttpRequests, XML / JS transformation and DOM manipulation
- Applying AJAX is **all** about
 - Responsiveness
 - Interactivity
 - Intuitiveness
 - Timing
 - Engagement



UE paraphrased

- Users don't care about how you do things
- Users care about how they can get things done
- We want to enhance the user experience because it makes sense from a business perspective



Agenda

- JSF and AJAX recap
- User experience
- **Why use AJAX frameworks?**
- Frameworks positioning
- Demonstration of framework usage
- Frameworks pros and cons
- Questions



AJAX development levels*

- Snippet
 - The odd bit of JavaScript in your pages
- Widget
 - New controls that are AJAX enabled
- Framework
 - Infrastructure for generic interaction amongst components
- Enhanced framework
 - Intelligent interaction between client **and** server

* Ray Valdes, Gartner



Why use AJAX frameworks?

- Transport en DOM manipulation coding is better left to specialists
- Provides a sensible abstraction level to interact
- Application scope and complexity will increase
- A good framework gets you 90% of what you need but can be circumvented when required
- Community / documentation



Agenda

- JSF and AJAX recap
- User experience
- Why use AJAX frameworks?
- **Frameworks positioning**
- Demonstration of framework usage
- Frameworks pros and cons
- Questions



Frameworks

- AJAX 4 JSF (JBoss) – 3.1.0 with RF
- RichFaces (JBoss) – 3.1.0
- ICEFaces (ICESoft) – 1.7.0-DR1

- Apache MyFaces 1.1.5
- Can be used with Facelets

- All frameworks are OS these days



AJAX for JSF (A4J)



- Adds AJAX capabilities to existing or new components
- Provides components, but almost no UI components (visible)
- Focusses on generic functionality and can be used in combination with many other component sets



RichFaces

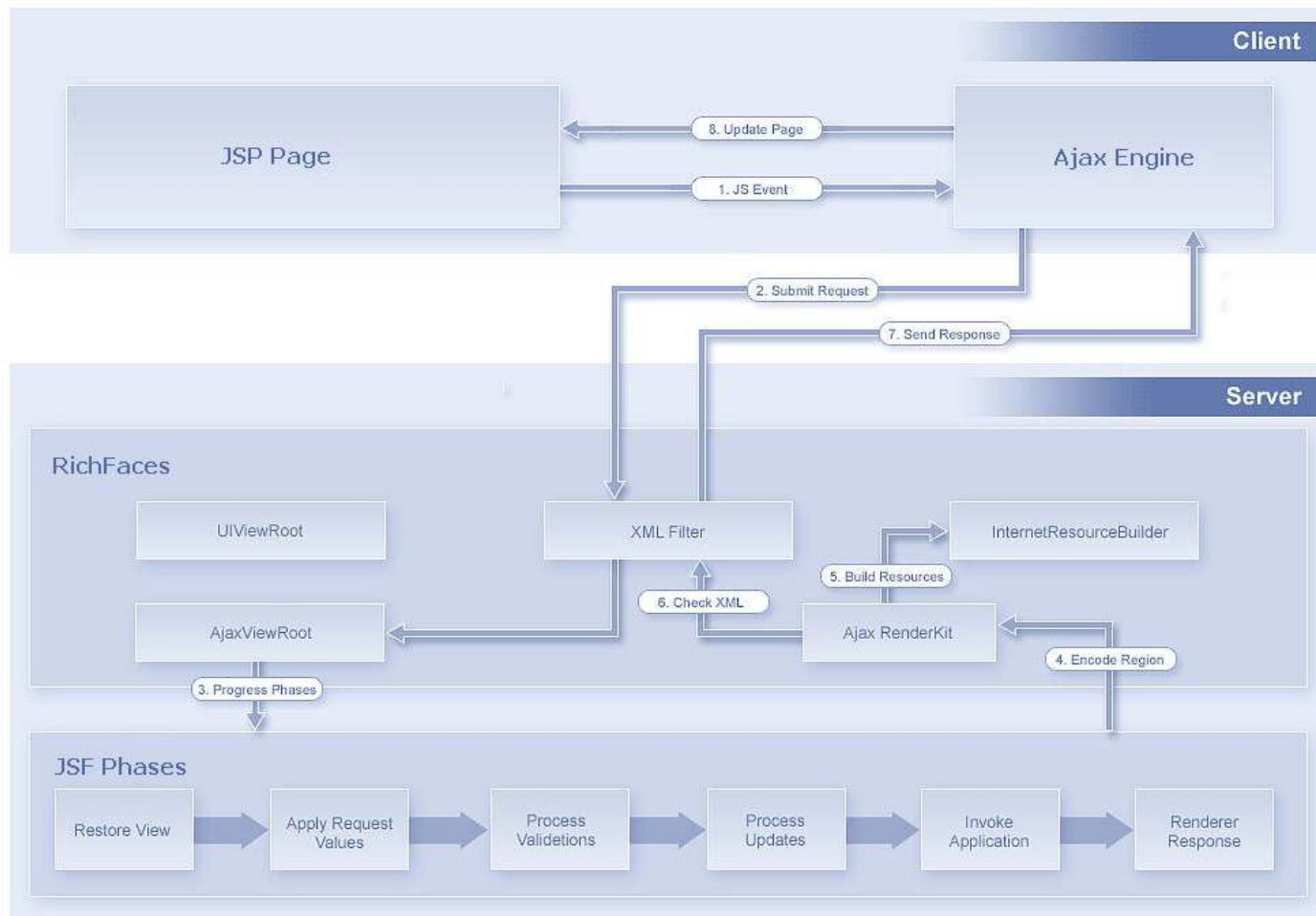


- Component set (55) of which many can be AJAX enabled by default
- Built on top of A4J
- Integrated into JSF lifecycle
- Supports PPR, Push style
- Easy skinning





RichFaces: Architecture





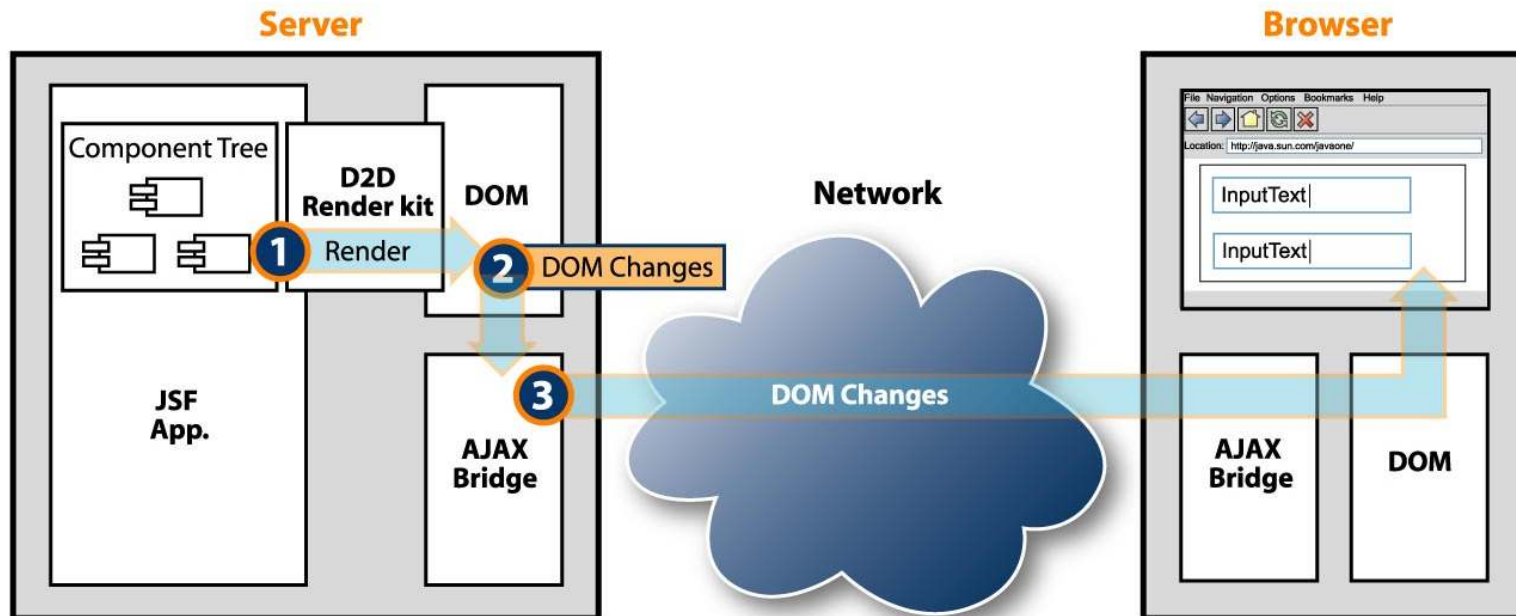
ICEFaces

- AJAX enabled component set
- Direct to DOM (D2D) rendering
 - Persistent server side DOM that components render to directly
- Incremental browser DOM update & SIR
 - Automatic determination of updated components based on the server side DOM
- Automated partial submit
 - Low level control of form submission that utilizes the JSF lifecycle intelligently





D2D rendering in ICEFaces





D2D rendering in ICEFaces



Agenda

- JSF and AJAX recap
- User experience
- Why use AJAX frameworks?
- Frameworks positioning
- **Demonstration of framework usage**
- Frameworks pros and cons
- Questions



Demonstration



GreenFinance
the green investors

Case: GreenFinance

- Preserving the planet one dollar at a time
- **Existing** customers call in to enquire about current products, new products and advice
- Customer support employees lookup data about customers and patch through to specialists
- Some services are subscribed to or terminated during conversations



Reference application



Areas of improvement

	Responsiveness	Interactivity	Intuitiveness	Timing	Engagement
Indicate and quantify activity	X			X	
Searchmode		X	X		
Productinformation		X	X		
Specialists' availability		X		X	X
Feedback adding / removing	X	X			X
Workflow		X	X	X	X

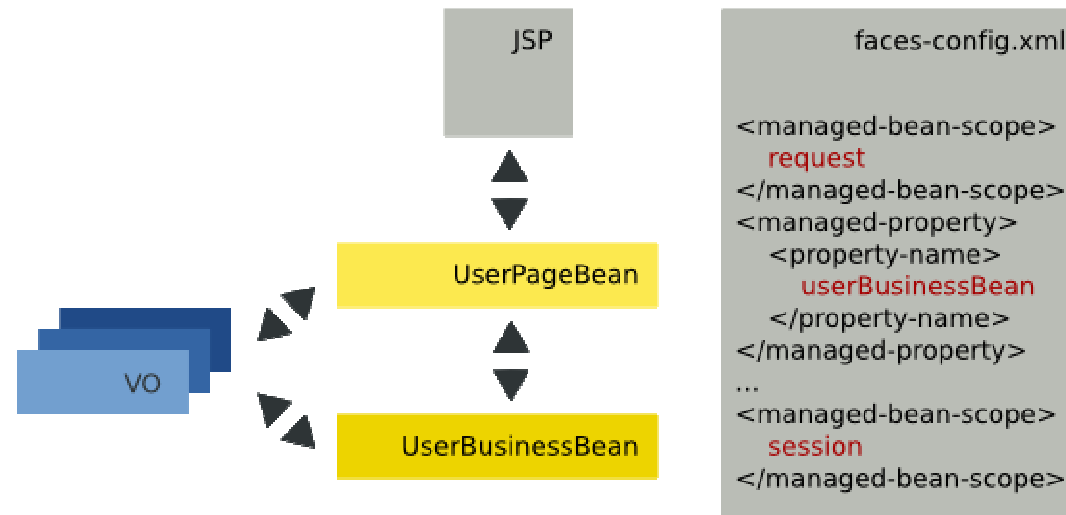


Demonstration RichFaces



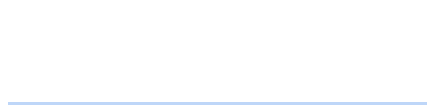
Demonstration: general setup

- We use the following bean convention:





RichFaces - Status



```
<a4j:status for="loader">
  <!--
    The image that shows the connection is active
    for any ajax call in the page -->
  <f:facet name="start">
    <h:graphicImage value="/images/ajax-loader.gif" />
  </f:facet>
  <f:facet name="stop">
    <!-- Status idle information could go here -->
  </f:facet>
</a4j:status>
```

```

<!-- Defines a set of tabs which can be switched using AJAX -->
<rich:tabPanel id="searchTabs" switchType="ajax" selectedTab="#{searchPB.searchMode}">

  <!-- The tab used to search by customer number -->
  <rich:tab
    name="customerNumber"
    label="#{searchBundle.searchByCustomerNumberHeading}"
    actionListener="#{searchPB.onSelectTab}"
    reRender="searchTabs"
    bypassUpdates="true">

    <h2>
      <h:outputText value="#{searchB
    </h2>

    <h:panelGrid columns="2">
      <h:outputText value="#{custome
      <h:panelGroup>
        <h:inputText id="customerNun

        </h:inputText>
        <h:message for="customerNi
      </h:panelGroup>

      <!-- A4j button makes the post as
    <a4j:commandButton
      value="#{searchBundle.labelSearchButton}"
      type="submit"
      actionListener="#{searchPB.searchCustomers}"
      reRender="searchResults" />

    </h:panelGrid>

  </rich:tab>

```

```

/**
 * Sets the search mode when the user selects a different tab.
 * @param event the ActionEvent that triggered the call
 */
public void onSelectTab(ActionEvent event) {
    HtmlTab tab = (HtmlTab)event.getComponent();
    String tabName = (String)tab.getName();
    //Required to comply with legacy implementation
    super.setSearchMode(tabName);
    try{
        Thread.sleep(2000);
    }catch (InterruptedException e){
    }
}

```



RichFaces – Search progress

Search by Client Number Search by Last Name / Date of Birth Se

Search by Last Name / Date of Birth

Last name:

Date of birth: 



```
<!-- Poller checks for searching progress while the search runs and updates the progress bar -->
<a4j:region id="pollerSearchProgress">
  <a4j:poll interval="500" reRender="searchProgressBar" action="#{searchProgressPB.getSearchProgress}" />
</a4j:region>

<h:panelGroup id="searchProgressBar">
  <h:panelGroup rendered="#{searchProgressPB.searchProgress > 0 and searchProgressPB.searchProgress < 100}">
    <div class="searchProgressBar" style="height: 20px; width: #{3 + (searchProgressPB.searchProgress * 0.5)}em;">
      <h:outputText value="#{searchProgressPB.searchProgress} %" />
    </div>
  </h:panelGroup>
</h:panelGroup>
```



RichFaces - Autosuggest

Home Start new call Search

```
<!-- Suggestionbox for autosuggest -->
```

Search clients and

```
public ArrayList<Customer> autocomplete(String pref) {
```

Search by Client Number

```
    ArrayList<Customer> result = new ArrayList<Customer>();  
    Collection<Customer> allCustomers = this.m_userBean.getCustomers();
```

Search by Last Name

Last name:

ij

Date of birth:

Dijkstra

Search

van Wij

Crujff

Rijker

```
    for (Customer customer : allCustomers) {  
        //adds if the last name contains the provided criteria.  
        if (customer.getLastName().toLowerCase().contains(pref.toLowerCase())) {  
            result.add(customer);  
        }  
    }
```

```
    return result;
```

```
}
```

Search result

```
</rich:suggestionbox>
```

Current call dura



RichFaces - Searching

```
<!-- A4J button makes the post as an AJAX call and rerenders the results on completion -->  
<a4j:commandButton  
  value="#{searchBundle.labelSearchButton}"  
  type="submit"  
  actionListener="#{searchPB.searchCustomers}"  
  reRender="searchResults" />
```

```
<h:panelGroup id="searchResults">  
  <h2>  
    <h:outputText value="#{searchBundle.resultsHeading} ({fn:length(searchPB.searchResults)})" />  
  </h2>  
  
  <h:dataTable  
    value="#{searchPB.searchResults}"  
    var="customer"  
    rowClasses="row1, row2"  
    styleClass="salessupport"  
    rendered="#{searchPB.showSearchResults}">  
  
    <h:column>  
      <f:facet name="header">
```



RichFaces - Workflow

```
<rich:panelBar id="workflowPanelBar" selectedPanel="#{customerPB.defaultPanel}" height="70%">
  <rich:panelBarItem name="personalia" label="#{richBundle.panelBarItemPersonaliaHeading}">
    <ui:include src="customerDetailsPersonalia.jspx" />
  </rich:panelBarItem>
  <rich:panelBarItem name="products" label="#{richBundle.panelBarItemProductsHeading}">
    <ui:include src="customerDetailsProducts.jspx" />
  </rich:panelBarItem>
  <rich:panelBarItem name="productInformation" label="#{richBundle.panelBarItemProductInformationHeading}">
    <ui:include src="customerDetailsProductInformation.jspx" />
  </rich:panelBarItem>
</rich:panelBar>
```

```
51
52 @Override
53 public void onViewProduct(ActionEvent event) {
54     super.onViewProduct(event);
55     HtmlPanelBar panelBar = (HtmlPanelBar)event.getComponent().findComponent(":workflowPanelBar");
56     panelBar.setSelectedPanel(RFCustomerPageBean.PANEL_PRODUCT_INFORMATION);
57 }
58
```



RichFaces - Specialists

Available products

↑ Personalia

ID	Name	Price		
32485	GreenCar WA	28.0		0
183187	Rechtsbijstand	14.0		0
123468		38.9		1

`<!-- Poller ensures the advisor availability remains up to date -->`

```
<a4j:region id="pollerCurrentProducts">
```

```
  <a4j:poll interval="5000" reRender="currentProducts" action="#{customerPB.getCurrentProducts}" />
```

```
</a4j:region>
```

```
<h:dataTable
```

```
  id="currentProducts"
```

```
  value="#{customerPB.currentProducts}"
```

```
  var="advisableProduct"
```

```
  rowClasses="row1, row2"
```

```
  styleClass="salessupport">
```



RichFaces - Feedback

Aansprakelijkheidsverzekering				34782	GreenTravel reisverzekering
Inboedelverzekering	38.9		1		
Brandverzekering	5.3		1	32482	GreenCar Allrisk

```
public void dropAvailableProduct(DropEvent event) {  
    Product selectedProduct = findSelectedProduct();
```

```
    super.getCustomerBusinessBean().addProduct(selectedProduct, getSelectedCustomer());  
    super.getCustomerBusinessBean().setProductDetails(false);
```

```
}
```

```
public void dropCurrentProduct(DropEvent event) {  
    Product selectedProduct = findSelectedProduct();
```

```
    super.getCustomerBusinessBean().removeProduct(selectedProduct, getSelectedCustomer());  
    super.getCustomerBusinessBean().setProductDetails(false);
```

```
}
```

```
private Product findSelectedProduct() {
```

```
    FacesContext facesContext = FacesContext.getCurrentInstance();
```

```
    String productId = facesContext.getExternalContext().getRequestParameterMap().get("productId").toString();
```

```
    Product selectedProducts = new Product();
```

```
    List<Product> allProducts = super.getCustomerBusinessBean().getProductApplicationBean().getProducts();
```

```
    for (Product product: allProducts) {
```

```
        if(product.getId().toString().equals(productId)) {  
            selectedProducts = product;
```

```
        }
```

```
    }
```

```
    return selectedProducts;
```

```
}
```



RichFaces - Offer

Offer

2 8 2

il
la
'
n
ir
te

Your reference: 123
Wilma Dijkstra - de G
 Boschdijk 170
 5613op Eindhoven

Offer for persons:

ID	
54722	Aansprak
123468	Inboedelv
123467	Brandver:
32485	GreenCar
183187	Rechtsbij:

This offer is valid for a pr

```

<!-- Toolbar can group any type of other components and order them -->
<rich:toolbar itemSeparator="grid">
  <rich:toolbarGroup location="left">
    <h:form>
      <!-- InputNumberSlider allows for inputting numbers by dragging on a scale -->
      <rich:inputNumberSlider
        value="#{offerPB.offerExpiration}"
        minValue="2"
        maxValue="8"
        step="2"
        enableManualInput="false">
        <!-- Adds AJAX support to a JavaScript DOM event -->
        <a4j:support event="onchange" reRender="offerExpiration" />
      </rich:inputNumberSlider>
    </h:form>
  </rich:toolbarGroup>
  <rich:toolbarGroup location="right">
    <h:graphicImage value="/images/icons/printer.png" />
  </rich:toolbarGroup>
</rich:toolbar>

```



Demonstration ICEFaces



ICEFaces – D2D Rendering

```
<!-- One big form allows for detailed updates because of ICEFaces' incremental DOM updates -->  
<ice:form>
```

```
  <div id="header">  
    <h:graphicImage value="/images/greenFinance.png" alt="#{supportBundle.applicationName}" />  
  </div>
```



ICEFaces - Status



```
<div id="timer">
  <h:outputText value="#{supportBundle.currentCallDuration}" rendered="#{genericPB.loggedIn}"/>

  <!-- Output the time the call has been running for -->
  <ice:outputText
    value="#{genericPB.timerValueAsString}"
    rendered="#{genericPB.loggedIn}"
    effect="#{genericPB.timerEffect}"/>

  <div class="ajaxLoader">
    <!-- Component auto detects connection status -->
    <ice:outputConnectionStatus/>
  </div>
</div>
```



ICEFaces - Tabs

Home Start new call Search

Search clients and products

Search by Client Number

Search by Last Name / Date of Birth

Search by Zip / Number

Search by Client Number

Client number:

Search

```

Cu.3 public class SearchTabChangeListener implements TabChangeListener {
    .4
    .5     private final static Log LOG = LoggerFactory.getLog(SearchTabChangeListener.class);
    .6
    <i>.7e public void processTabChange(TabChangeEvent event) throws AbortProcessingException {
    .8         LOG.info("Tab change event" + event.getNewTabIndex());
    .9
    ?0         //This listener isn't a managed bean, so it needs to get the businessbean through the context
    ?1         FacesContext context = FacesContext.getCurrentInstance();
    ?2         SearchBusinessBean sb = (SearchBusinessBean) context.getExternalContext().getSessionMap().get("searchBB");
    ?3
    ?4         //Method provided by TabChangeEvent
    ?5         int index = event.getNewTabIndex();
    ?6
    ?7         //The legacy search business bean requires this to be set before searching
    ?8         if (index == 1) {
    ?9             sb.setCurrentSearchMode(SearchBusinessBean.LAST_NAME_BIRTH_DATE);
    ?0         } else if (index == 2) {
    ?1             sb.setCurrentSearchMode(SearchBusinessBean.ZIP_NUMBER);
    ?2         } else {
    ?3             sb.setCurrentSearchMode(SearchBusinessBean.CUSTOMER_NUMBER);
    ?4         }
    ?5     }
    ?6 }
```

tener"/>



ICEFaces - Autosuggest

```
/**
 * Called when a user has modified the SelectInputText value. This method
 * call causes the match list to be updated.
 *
 * @param newValue the new value
 */
public void autocomplete(String newValue, int maxMatches) {
    List<SelectItem> matchList = new ArrayList<SelectItem>(maxMatches);
    Collection<LastName> allLastNames = this.m_userBean.getLastNames();


    for (LastName lastName : allLastNames) {
        //adds if the last name contains the provided criteria.
        int counter = 0;
        if (lastName.getLastName().toLowerCase().contains(newValue.toLowerCase())) {
            if (counter > maxMatches) break;
            matchList.add(new SelectItem(lastName, lastName.getLastName()));
            counter++;
        }
    }
    // assign new matchList
    if (this.m_matchesList != null) {
        this.m_matchesList.clear();
        this.m_matchesList = null;
    }
    this.m_matchesList = matchList;
}

public List<SelectItem> getMatchesList() {
    return m_matchesList;
}
}
```

String();



ICEFaces – Search progress

Date of birth: 

Current call duration: 00:50

```
<ice:commandButton value="#{searchBundle.labelSearchButton}"
  type="submit"
  disabled="#{searchProgressRB.disableSearchButton}"
  actionListener="#{searchPB.searchCustomersWithProgress}"/>
```

```
<!-- Indicates the search progress -->
<!-- There is no reRender property here! -->
<ice:outputProgress
  value="#{searchProgressRB.percent}"
  label="Zoeken..."
  labelComplete="100%"
  labelPosition="embed"
  binding="#{searchProgressRB.searchProgressBar}"
  rendered="#{searchProgressRB.disableSearchButton}" />
```



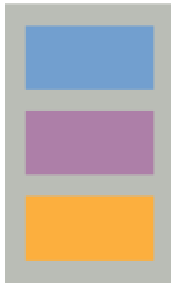
ICEFaces - Specialists

```
6
7 public class IFUserApplicationBean extends UserApplicationBean {
8
9     public static final String AVAILABLE_ADVISORS_RENDERER = "AVAILABLE_ADVISORS_RENDERER";
10
11     private RenderManager m_renderManager;
12     /**
13      * This method is called from the application bean where the availability change is handled.
14      * If the availability of any advisor has changed, the renderer will make sure we see updates
15      */
16     @Override
17     public void afterAvailabilityChange() {
18         super.afterAvailabilityChange();
19         if (m_renderManager != null) {
20             OnDemandRenderer advisors = m_renderManager.getOnDemandRenderer(AVAILABLE_ADVISORS_RENDERER);
21             if (advisors != null) {
22                 advisors.requestRender();
23             }
24         }
25     }
26
27     public RenderManager getRenderManager() {
28         return m_renderManager;
29     }
30
31     public void setRenderManager(RenderManager renderManager) {
32         m_renderManager = renderManager;
33     }
34 }
```

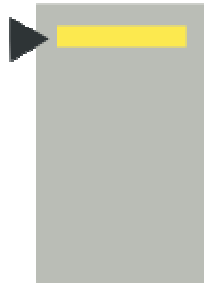


ICEFaces - RenderHub

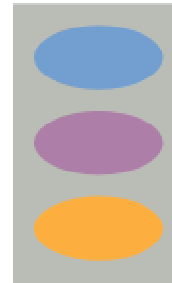
Component State



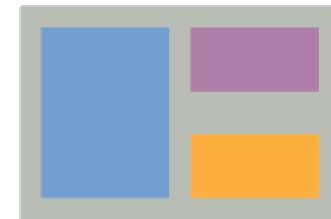
RenderHub Queue



Serverside DOM



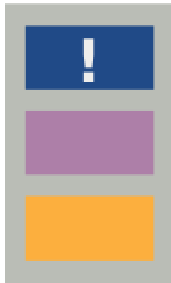
Screen



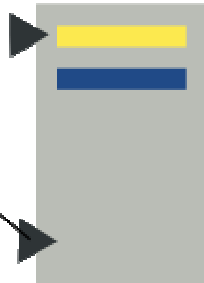


ICEFaces - RenderHub

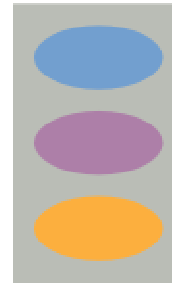
Component State



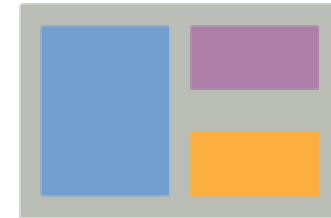
RenderHub Queue



Serverside DOM



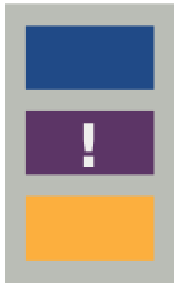
Screen





ICEFaces - RenderHub

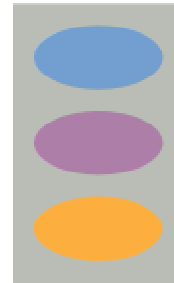
Component State



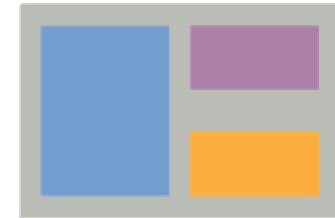
RenderHub Queue



Serverside DOM



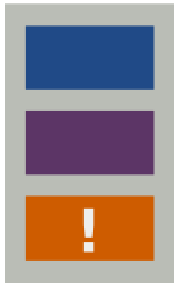
Screen





ICEFaces - RenderHub

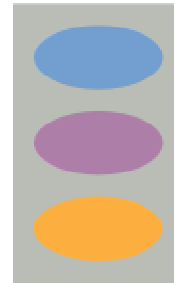
Component State



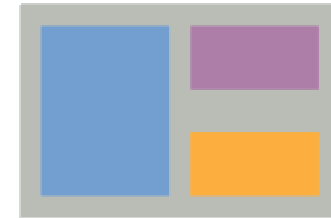
RenderHub Queue



Serverside DOM



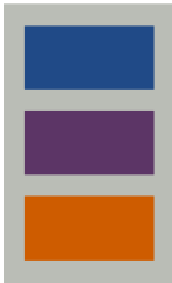
Screen



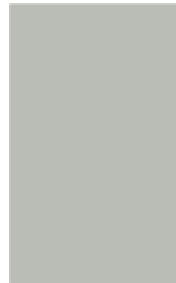


ICEFaces - RenderHub

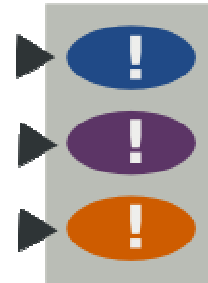
Component State



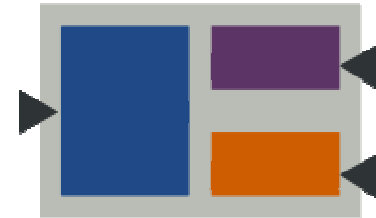
RenderHub Queue



Serverside DOM



Screen





Agenda

- JSF and AJAX recap
- User experience
- Why use AJAX frameworks?
- Frameworks positioning
- Demonstration of framework usage
- **Frameworks pros and cons**
- Questions



Frameworks: RichFaces Pro

- Components UI is consistent, customizable, clean and modern.
- Why reinvent the wheel with A4J
- Datatables provide extensive functionality
- Implementation is easy as a-b-c
- Non-intrusive approach
- Ideal for step-by-step integration



Frameworks: RichFaces Cons

- Documentation: can you say scrollwheel?
- Project needs maturity
- A4J is supplied
- JS bloat with component increase on page



Frameworks: ICEFaces Pros

- Push technology can be complex, but is powerful once you get the hang of it
- D2D rendering combined with incremental update
- Broad component functionality
- No JS coding, lightweight AJAX bridge doesn't bloat with complexity increase
- High responsiveness



Frameworks: ICEFaces Cons

- Implementation is fragile and far from trivial
- Intrusive architecture
- Component default style
- Session scoped beans yield unexpected results
- Documentation is often incomplete



Frameworks: General remarks

- Implementation is not as trivial as you might hope for
- You **will** need to append or edit your core functionality (e.g. beans) to incorporate frameworks
- Documentation & showcase often shows too simple examples
- Style mixups might prove troublesome with existing applications



Implementation effort

	RichFaces / A4J	ICEFaces
Lines of code	371	635
Methods	19	27
Implementation	2 hours	1½ days
Development (est.)	16 hours	20 hours



Choosing a framework

- Evaluate your current application ecosystem
- Consider community vibrance, development activity and support options
- Study documentation and don't be misled by examples: your results most likely won't be examples either
- Unique features you may require

.nl.
jug



Questions

- Any questions?
- Interested in source code? Send me an email!

tiese.barrell@atosorigin.com

Atos Origin
Emerging Technologies







ICEFaces support

- Application Server Support:
 - Apache Tomcat
 - BEA Weblogic Server
 - JBoss Application Server
 - IBM Websphere Application Server
 - Oracle Application Server Container for J2EE (OC4J)
 - SAP NetWeaver
 - Sun GlassFish
 - Sun Java System Application Server
 - Webtide Jetty
- Java EE Portals and Frameworks:
 - Liferay Portal 4.3
 - JBoss Seam 1.3
 - JavaServer Faces (JSF) 1.1, 1.2,
 - Facelets
 - Spring Web Flow (early integration)
- Supported Web Browsers:
 - Internet Explorer 6, 7
 - Firefox 1.x, 2.0
 - Mozilla 1.7.x and Netscape 7.x
 - Safari 1.3
 - Opera 9.x, Opera Mobile (preliminary)



RichFaces Support

- Supported JavaServer Faces Implementations
 - Sun JSF 1.1 RI - 1.2, MyFaces 1.1.1 - 1.1.5, Facelets JSF 1.1.1 - 1.2
- Supported Servers
 - Apache Tomcat 4.1 - 6.0
 - IBM WebSphere 5.1 - 6.0
 - BEA WebLogic 8.1 - 9.0
 - Oracle AS/OC4J 10.1.3
 - Resin 3.0
 - Jetty 5.1.X
 - Sun Application Server 8 (J2EE 1.4)
 - Glassfish (J2EE 5)
 - JBoss 3.2 - 4.0.x
 - Sybase EAServer 6.0.1
- Supported Browsers
 - Internet Explorer 5.5 - 7.0, Firefox 1.5 - 2.0, Opera 8.5 - 9.0, Netscape 7.0, Safari 2.0