



ICEfaces.org JavaOne

Assembling Ajax Applications With Power Tools

Ted Goddard, Ph.D.

Senior Software Architect
ICEsoft Technologies, Inc.
<http://www.icefaces.org>

TS-9517

Assembling Ajax Applications

With power tools

Learn how to quickly develop Ajax applications using a powerful framework (Project jMaki, Project Dynamic Faces, or ICEfaces) and IDE design features.

Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion



DEMO

AuctionMonitor Demo



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion

Ajax Without Frameworks or Tools

Keep first-aid kit on hand

- Develop in JavaScript™ programming language?
 - Fashionable, but still not a good idea
- Write Servlets?
 - Not even fashionable
- Will it be reliable and secure?
- Ajax requires low-level code easily available in frameworks and tools; Why re-implement it?

Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion



Project jMaki

Universal components

- Project jMaki provides a universal tag library for Ajax components
 - Dojo, Flickr, Google, Spry, script.aculo.us, Yahoo

```
<a:ajax name="dojo.combobox" service="comboBoxData.js" />
```

- Ajax components communicate within the page in JavaScript programming language via Project jMaki glue

```
jmaki.publish("/yahoo/geocoder", data.coordinates);
```

- Install Project jMaki NetBeans™ software plug-in

```
https://ajax.dev.java.net/jmaki-plugin.html
```


Project jMaki Development



Process data with JSP technology

- Bind to a JavaServer Pages™ (JSP™) file that accepts “value” parameter

```
<a:ajax name="scriptaculous.inplace"  
  value="Click to Test"  
  service="inplaceService.jsp" />
```

- inplaceService.jsp returns String or JSON data, not HTML
- Operate on application beans

```
<% out.print(  
  request.getParameter("value").length() );  
%>
```

Project jMaki Development

Glue components with publish/subscribe



```
<a:ajax name="dojo.fisheye"
  args="{topic : '/dojo/fe'}"
  value="[
    {iconSrc:'icon.jpg',caption:'jMaki', index: 1},
  ]"
/>
```

```
<script >
  function fisheyeListener(item) {
    var caption = item.target.caption;
    document.getElementById("targetdiv")
      .innerHTML = caption;
  }

  jmaki.subscribe("/dojo/fe", fisheyeListener);
</script>
```



Project jMaki Design

Ajax component tags

- Page design with pure JSP or JavaServer™ Faces technology
- Embed Project jMaki component tags

```
<%@ taglib prefix="a" uri="http://java.sun.com/jmaki" %>
```

```
<html>
```

```
  <h2>Script.aculo.us In Place Editor Test</h2>
```

```
  <a href="index.jsp">Back to Samples</a>
```

```
  <hr>
```

```
  <a:ajax name="scriptaculous.inplace"
```

```
    value="Click" service="inplaceService.jsp"/>
```

```
  <br>
```

```
</html>
```



DEMO

Project jMaki with NetBeans Software



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion

Project Dynamic Faces

Update zones

- Project Dynamic Faces allows Ajax update zones to be applied to pages with existing components
 - Specified events execute the application and update content in the zones
 - Specify events with `DynaFaces.fireAjaxTransaction`
 - Or specify entire regions with `jsfExt:ajaxZone`

```
<jsfExt:ajaxZone id="zone2"  
  action="#{carstore.currentModel.updatePricing}">
```

```
<!-- JSF and HTML content goes here. -->
```

```
</jsfExt:ajaxZone>
```

Project Dynamic Faces Development

Subtree execution and rendering

- Dispatch Ajax events to execute and render specified component subtrees

```
<jsfExt:scripts />
```

```
<h:inputText id="fieldSearch" />
```

```
<h:commandButton id="buttonSearch" value="Search"  
    onclick="DynaFaces.fireAjaxTransaction(this,  
        { execute:'buttonSearch,fieldSearch',  
          render:'panelResults,panelSummary' } );  
    return false;"  
    actionListener="#{bean.search}" />
```

Project Dynamic Faces Design



Associating events with zones

- Develop standard JavaServer Faces application
- Assign ids to significant panels and components
- Wrap regions in `jsfExt:ajaxZone` if input and output regions are simple
- Or add `DynaFaces.fireAjaxTransaction` event handlers
 - If regions are complex
 - If various JavaScript programming language events should trigger Ajax



DEMO

Project Dynamic Faces with NetBeans Software



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion



ICEfaces

Transparent Ajax

- ICEfaces renders JavaServer Faces components as Ajax components
- ICEfaces determines minimal page update

```
<h:inputText value="#{bean.text}" />
```

- Includes rich components and extensions

```
<h:panelGroup draggable="true" />
```

- Provides Ajax push for collaborative applications
 - Incrementally update the page from the server



ICEfaces Philosophy

The language is the power tool

```
public class PageBean {
    String text;

    public String getText(){
        return text;
    }

    public void setText(String text){
        this.text = text;
    }
}
```

PageBean.java

```
<f:view
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html">
<html>
    <body>

        <h:form>
            <h:inputText
                value="#{pageBean.text}"/>
        </h:form>

    </body>
</html>
</f:view>
```

Page.xhtml

A language for Ajax Push that preserves Designer and Developer roles

ICEfaces Development



I want my MVC

- The JavaBeans™ architecture model determines all component relationships
 - Developer focuses on Java™ code, allowing designer to focus on pages
- Integration with JBoss Seam for conversation scope and annotation-managed beans
- Ajax push requires no model or page changes
 - Simply call `PersistentFacesState.render()` and ICEfaces will update the page
 - Use a `RenderManager` for thread pooling



ICEfaces Design

Transparent Ajax

- Designer-friendly pages
- Pages contain no JavaScript programming language and no event/component wiring
- No page changes for Ajax push
- Pages are purely dynamically bound XHTML (Facelets) or JavaServer Faces components/JSP page
- ICEfaces component theme customized with a structured CSS file
- Tree, Tabs, Calendar, Drag and Drop, and Popup panel components



DEMO

ICEfaces with NetBeans Software



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion

Ajax Security

As secure as web1.0?

- Ajax security adds nothing new on the wire
- But different Ajax practices present different attack surfaces into your application
- Ajax attacks can be less visible to the user
 - Browser-based scripted attacks using XMLHttpRequest
- Application logic in JavaScript programming language is open to attack
- Dynamically assembled JavaScript programming language difficult to audit

Ajax Security Specifics

Security is framework dependent

- Project jMaki
 - Service bindings subject to attacks and script injection
 - Business logic in JavaScript programming language subject to attack
- Project Dynamic Faces, ICEfaces
 - JavaServer Faces component tree processes HTML form input so the attack surface is just the user interface
 - Make use of default escaping in `outputText`
 - Resists JavaScript programming language cross-site scripting
 - Use high-level database API (like Hibernate)
 - Hand-coded SQL risks SQL-injection attacks

Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion



Ajax Push

Application-initiated page updates

- Push with ICEfaces requires no view changes
 - Keep your .jsp, .jspx, .xhtml intact
- Determine update triggers and call render() on the server or use a RenderManager
 - New chat text, new bid, slide change, time elapsed
- AKA “Comet” (Dojo), “Reverse Ajax” (DWR)
- Web applications for the participation age, allowing people to communicate and collaborate

Protocol for Ajax Push

GET /auctionMonitor/block/receive-updates?icefacesID=1209765435 HTTP/1.1

Accept: */*

Cookie: JSESSIONID=75CF2BF3E03F0F9C6D2E8EFE1A6884F4

Connection: keep-alive

Host: vorlon.ice:18080



Chat message "Howdy"

HTTP/1.1 200 OK

Content-Type: text/xml;charset=UTF-8

Content-Length: 180

Date: Thu, 27 Apr 2006 16:45:25 GMT

Server: Apache-Coyote/1.1

```
<updates>
```

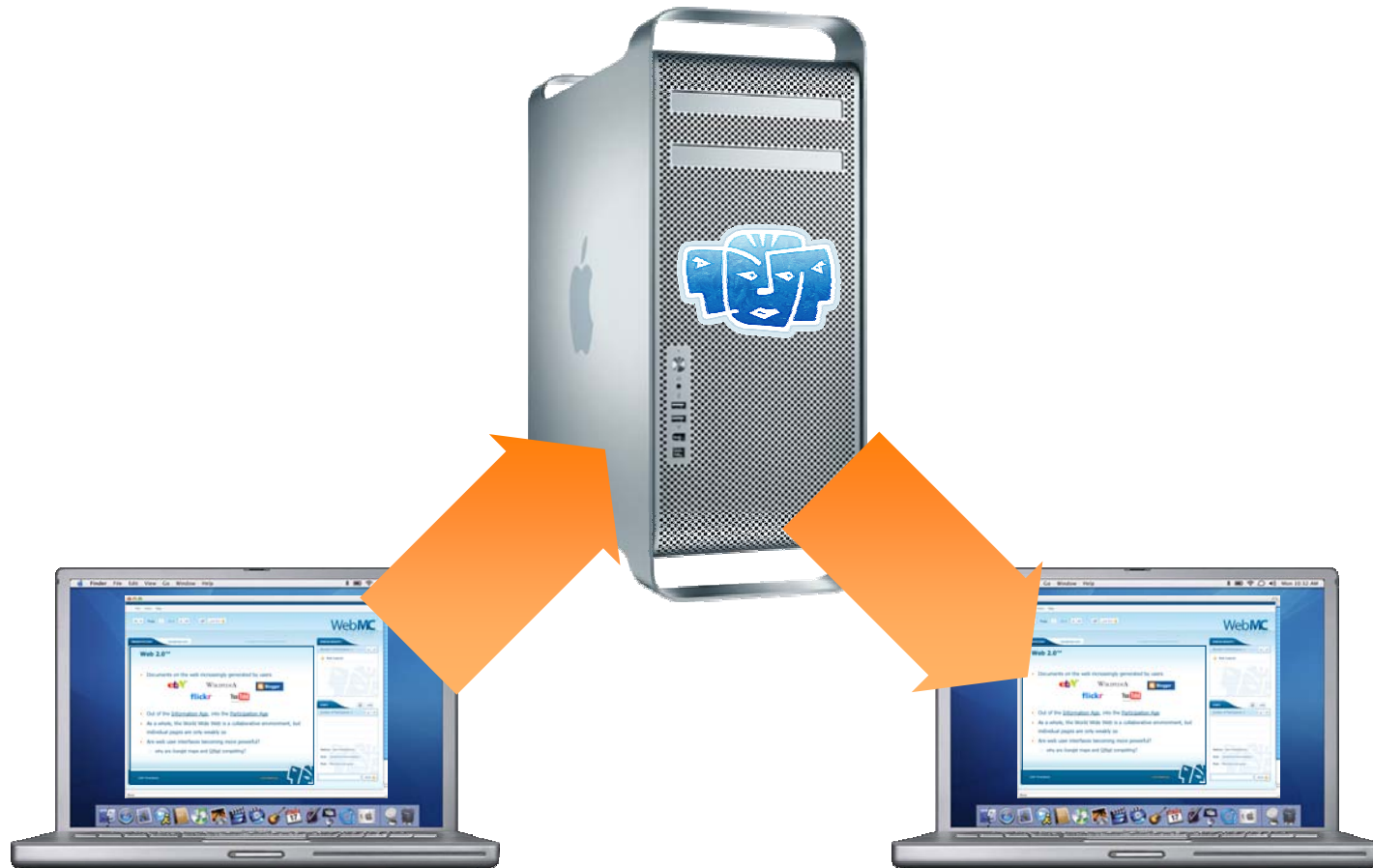
```
  <update address="_id0:_id5:0:chatText">
```

```
    <span id="_id0:_id5:0:chatText">Howdy</span>
```

```
  </update>
```

```
</updates>
```

Server-Mediated Collaboration



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion

Facelets

XHTML templates

- XHTML pages with Tapestry-like syntax
- Develop with standard JavaServer Faces managed beans
- Design with HTML editor
- Tag definition and template insert, composition, fragment, and decorate for effective re-use
- More efficient, does not require JSP technology engine

Facelets page.xhtml

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

```
<form jsfc="h:form">
```

```
<img jsfc="h:graphicImage" value="#{user.slideURL}"/>
```

```
<div id="chatDiv" >
```

```
Chat
```

```
#{user.chatLog}
```

```
<input type="text" jsfc="h:inputText"
      value="#{user.chatInput}" />
```

```
<input type="submit" jsfc="h:commandButton"
      actionListener="#{user.submit}" />
```

```
</div>
```

```
</form>
```

Design and Development

Lessons from designer/developer collaboration

- CSS, XHTML, and Java code are distinct
 - Style, layout, and logic require distinct expertise
 - Stylesheets more maintainable than inline styles
- Facelets allows designers to work with familiar tools throughout application cycle
- ICEfaces adds Ajax without disrupting design
- Don't try for pixel-perfect cross-browser design
- Encourage developers and designers to share actual code, screenshots, and console logs not just problem descriptions



DEMO

Facelets with Dreamweaver



Agenda

Ajax Without Frameworks/Tools?

Project jMaki: Universal Components

Project Dynamic Faces: Update Zones

ICEfaces: Transparent Ajax

Ajax Security: Trust Your Layers

Ajax Push: Collaborative Web

Facelets: for Designers and Developers

Conclusion

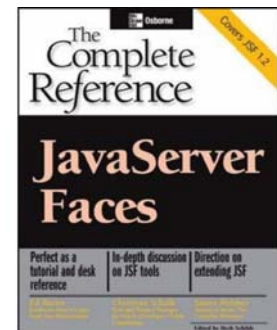
Summary

- Ajax without framework: assembly language
- Project jMaki: for adding JavaScript programming language Ajax components
- Project Dynamic Faces: for update zones
- ICEfaces: for Transparent Ajax, Ajax Push
- Facelets: for more power than JSP software
- NetBeans software: for Ajax development in any style

For More Information

- ICEfaces:
<http://www.icefaces.org>
- Project Dynamic Faces:
<http://jsf-extensions.dev.java.net>
- Project jMaki:
<http://ajax.dev.java.net>

- *Ajax in Practice*
- *JavaServer Faces:
The Complete Reference*





Q&A

TedGoddard@icesoft.com

ICESoft Technologies, Inc.
<http://www.icesoft.com>

<http://www.icefaces.org>



ICEfaces.org **JavaOne**

Assembling Ajax Applications With Power Tools

Ted Goddard, Ph.D.

Senior Software Architect
ICEsoft Technologies, Inc.
<http://www.icefaces.org>

TS-9517