



the
POWER
of
JAVA™



JavaServer™ Faces, AJAX and Portlets: Developing Apps Without Distress

Brendan Murray

Software Architect
IBM

<http://www.ibm.com>

TS-6877

Why Are We Here Today?

To Learn ...

How AJAX techniques can be used with
JavaServer™ Faces technology in a Portal
environment

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

Introduction and Background

What do we want to do?

- Enhance the application
 - Eliminate page reload
 - Increase speed/responsiveness
 - Enhance usability
 - Make application more interactive

Introduction and Background

What are the technologies?

- **JavaServer Faces technology**
 - Java Community Process™ (JCP) services JSR-127 (v1.2 is JSR-252)
 - Newest version 1.2 leverages updates in Java™ Platform, Enterprise Edition 5 (Java EE 5)
 - Widely supported: Sun, IBM, BEA, Oracle, etc.
 - Open Source versions:
 - Sun's implementation
 - Apache MyFaces
 - Provides a server-based GUI
 - MVC-2 architecture

Introduction and Background

What are the technologies?

- JavaServer Faces Technology
- **Portal and Portlets**
 - JSR-168 (v2.0 is JSR-286)
 - Broad support: Sun, IBM, BEA, Oracle, etc.
 - Open Source version: Apache Pluto
 - Provides an aggregation of web pages on the screen

Introduction and Background

What are the technologies?

- JavaServer Faces technology
- Portal and portlets
- **AJAX**
 - Originally called Remote Scripting
 - **Asynchronous JavaScript™ And XML**: snappy acronym coined by Jesse James Garrett, Adaptive Path
 - Widespread use triggered by Google's applications
 - Technique to create interactive web applications
 - Zero-footprint rich internet applications

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

The Challenges

What can JavaServer Faces technology do?

- Provides server-side GUI
- Provides server-side navigation
- Has an explicit request-response lifecycle
- Has no awareness of client-side behaviors

The Challenges

What can Portal/Portlets do?

- Multiple portlets displayed simultaneously
- A portlet has an explicit lifecycle
- Portlets can communicate with each other via the server
- Portal has no awareness of client-side behaviors
- Namespace encoding restricts the use of dynamically named objects

The Challenges

What can AJAX do?

- Assumes the entire application is in the client
- Can break synchronization of client and server
- No awareness of server-side behaviors

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

What Has to be Done

What do we have?

- Typical portal application
 - The portlet uses JavaServer Faces technology for GUI
 - Multiple submits to achieve basic functions
 - Entire portal page refreshed
 - All portlets redrawn

What Has to be Done

What do we have to do?

- Apply AJAX behaviors to the application
 - Retrieve data from JavaServer Faces technology lifecycle to update UI
 - Manage request to retrieve limited updates
 - Extend and simplify the UI using servlet data
 - For example, typeahead functions on input fields

What Has to be Done

What techniques can be used?

- Two main approaches
 - Request processed using full JavaServer Faces technology lifecycle
 - Eliminates most of the page's components
 - Pre-formatted markup returned to page
 - Request processed using partial JavaServer Faces technology lifecycle
 - Interrupted by phase listener
 - Request processed by special servlet
 - Separate data stream from JSF lifecycle
 - Request-response cycle independent of JSF
 - Stateless servlet

What Has to be Done

What exactly are the changes needed?

- JavaServer Faces technology
 - Lifecycle must manage an AJAX “submit”
 - One of two path taken
 - Render phase renders only limited components
 - Phase listener responds to request and ends cycle
- Extended functionality
 - Apply AJAX functionality to existing component
 - Retrieve data from special servlet

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

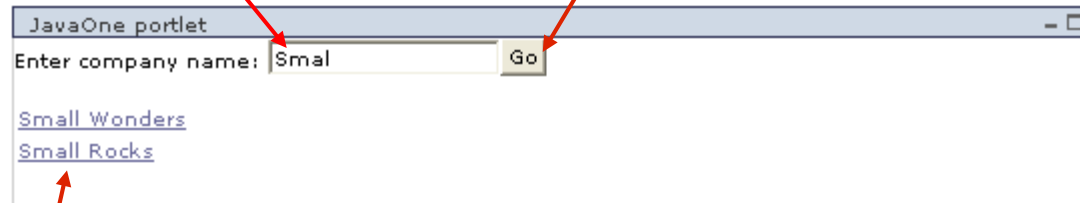
Summary

Typical Example

Traditional application flow—search

Enter search phrase

Click button to submit request



Response displays the search results

Typical Example

Traditional application flow—details portlet

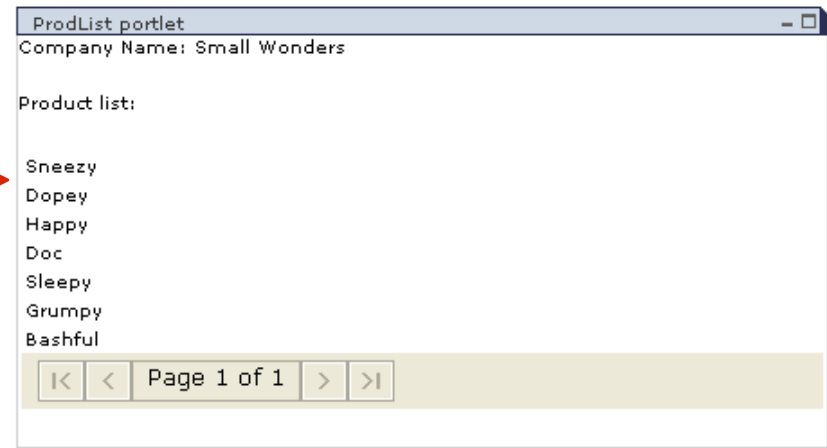
Click on a returned name



Continues usual processing



Loads information on the selected item



Typical Example

AJAX—enabling search portlet

- Enable typeahead on search phrase
 - While typing a narrowing list of results is displayed
 - Retrieves list dynamically from server
- Button no longer needed to submit request
 - Resolved to a single item in the list
 - Automatically trigger action based on the final selection
- Update occur in background
 - Page update *in situ*

Typical Example

AJAX—implementation strategies

Retrieve data from a servlet

- Doesn't participate in JavaServer Faces technology lifecycle
- No portal overhead
- Full control of page content
- Request processed by JavaServer Faces technology
 - Response from phase listener
 - Generate data and terminate lifecycle
 - Response from render phase
 - Only render what is required from the page
 - Filter out unnecessary markup
 - Skins and Themes
 - Other portlet content

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

Demo Contents

What we'll cover

- Track current request-response behaviors
- Add typeahead functionality to input field
- Retrieve data from three sources
 - External servlet
 - JavaServer Faces technology phase listener
 - Limited rendering of components

Demo Contents

Basic information

- Uses a simple bean for data
- Contains simple servlet as data source
- Contains portlet with phase listener
- Contains portlet with filtered rendering

Demo Contents

Typeahead functionality

- We want to use unchanged components
 - Typeahead added as a sub-component
 - This simply delivers JavaScript™ technology connection management
 - Data retrieved from
 - Stateless servlet
 - Portlet markup

Demo Contents

Typeahead functionality—markup

```
<P>Enter company name:
```

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

```
<hx:commandExButton ... />
```

```
<h:dataTable value="{myBean.words}"
              var="varwords">
  <h:column id="column1">
    <h:commandLink id="link1">
      <h:outputText value="{varwords.text}" />
    </h:commandLink>
  </h:column>
</h:dataTable>
```

Demo Contents

Typeahead functionality—markup

`<P>Enter company name:`

```
<h:inputText id="searchString"
    onblur="return func_1(this, event);">
    <hx:inputTypeahead value="{pc_page.myBean}">
</h:inputText>
```

- No longer need as much markup
 - Submit button replaced by onblur action
 - The table of links no longer needed
- New component as child decoration

Demo Contents

Retrieve page fragment from JavaServer Faces technology

- We need to limit the JavaServer Faces technology lifecycle
- Components tagged as needed
- Everything else on the page is ignored
- The markup can be processed in two ways
 - As part of the data returned by XMLHttpRequest
 - Extracted from data returned in a hidden iFrame

DEMO

Adding AJAX support to portlet applications

Agenda

Introduction and Background

The Challenges

What Has to be Done

Typical Example

Demo Contents

Demo

Summary

Summary

- AJAX, JSF and portlets work well together
- Even in portlets, AJAX can get data from servlet
- The true complexity is in limiting response size
- The real success of JavaServer Faces technology and AJAX is
 - AJAX solutions can be encapsulated in components
 - The complexity is hidden from developers

For More Information

- Sessions & BOFs
 - TS-1161—Evolving JavaServer™ Faces Technology: AJAX Done Right
 - TS-3187—Advanced JavaServer™ Faces Custom Component Development
 - BOF-2594—Java™ BluePrints Application: Creating AJAX-Enabled Web 2.0 Java Pet Store Demo Software
- Web sites
 - Ajax: A New Approach to Web Applications
 - <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- Books
 - JavaServer Faces: The Complete Reference - Ed Burns and Chris Schalk (Osborne)
 - Pro JSF and AJAX—Jonas Jacobi & John Fallows (APress)

Q&A

<code />



the
POWER
of
JAVA™



JavaOne
Part of the Network and Business Solutions

JavaServer™ Faces, AJAX and Portlets: Developing Apps Without Distress

Brendan Murray

Software Architect
IBM

<http://www.ibm.com>

TS-6877