Portlets (JSR-168)

Dave Landers
BEA Systems, Inc.
dave.landers@4dv.net
dave.landers@bea.com
Agenda

- Introduction
  - Concepts – Portals, Portlets, WebApps
- The Basics
  - API, Modes, States, Lifecycle of a Portlet
- More Detail
  - Names, URLs, Preferences, etc.
- Portlets and Servlets and JSPs
- Other Topics
- Examples
Introduction

Portals, Portlets, and all that
What Is a Portal?

- Portal is a Web Page
  - Hosted by a WebApp
  - Aggregation of Content
    - Portlets
    - And other things
      - Header
      - Footer
      - Menus
      - Links
  - There is no Portal spec
What Is a Portlet?

- Content for a Portal
- Mini-Application
- Portlets are assembled into a full Web page by the Portal

Portlets are just fragments of markup (HTML)
  - Avoid HEAD, TITLE, BODY, FRAME, etc.
Portal Page Elements

- Portal Page
- Portlet Content (Fragment)
- Portlet Window
- Decorations and Controls
Portlets! Don't! Have! To! Look! Like! Yahoo!
Portal Applications and WebApps

- Portal Application is part of a WebApp
  - Portal is hosted by WebApp
  - WebApp can also have “normal” Servlets and JSPs
  - Portlets deployed like Servlets
    - In WEB-INF/classes or WEB-INF/lib
    - Deployment Descriptor WEB-INF/portlet.xml

- Portlets can use Servlets and JSPs
  - To generate markup fragments or perform tasks

- Portlets have access to other Servlet container services
  - And J2EE services if WebApp in J2EE container
The Basics

Window States, Portlet Modes, Lifecycle, API
Portlet Window States

How the Portlet is rendered in the Portal

- NORMAL
  - Normal – what else to say?

- MINIMIZED
  - Portlet content is not rendered
  - Portal may still draw Portlet's decorations

- MAXIMIZED
  - Portlet has the whole Portal page (or a large portion of it)

- Custom States
  - Declared by Portlet, only used if supported by Portal
  - Example: HALF_PAGE, NARROW, FLOATING, WIDE ...
Portlet Modes

What kind of function the Portlet is performing

- **VIEW**
  - The normal mode – show your stuff

- **EDIT**
  - Changing preferences or properties – usually forms

- **HELP**
  - Should be useful or informative...

- **Custom Modes**
  - Declared by Portlet, only used if supported by Portal
  - “Standard” custom modes suggested by spec: ABOUT, CONFIG, EDIT_DEFAULTS, PREVIEW, PRINT
The Portlet API – javax.portlet

Portlet

- `processAction(ActionRequest, ActionResponse)`
  - The "action" phase (respond to action events)
- `render(RenderRequest, RenderResponse)`
  - The "render" phase (generate markup)

GenericPortlet

- `render()` forwards to mode-specific methods
  - `doView()`
  - `doEdit()`
  - `doHelp()`
  - `doDispatch()` can add processing for custom modes
Portlet Request Lifecycle

On any HTTP request, the Portal will invoke...

- **processAction()**
  - For zero or one Portlet
  - Only if request is an Action URL and "Targeted" to the Portlet

- **render()**
  - For all Portlets it decides should be rendered
  - No guarantees of order, *etc.*
  - Might run concurrently in separate Threads
  - Might run distributed on different VMs
  - Portlet returns content fragment

Portal assembles fragments into a web page
Lifecycle Implications

- Change things only in the processAction method
  - Preferences
  - Modes, State
  - Session Attributes
  - Changes will take effect for that Portlet's render phase
  - Can pass parameters to render via ActionResponse
  - Do not render any content from processAction

- Render should just construct HTML
  - Can build forms, links using Action URLs
    - Action URL targets the request back to processAction
Other javax.portlet APIs

Mostly, these match their counterparts in Servlet

- PortletRequest, PortletResponse
  - ActionRequest, ActionResponse
  - RenderRequest, RenderResponse
- PortletConfig
- PortletContext
- PortletSession
- PortletRequestDispatcher
- PortletPreferences
Code Break

HelloWorld Portlet

- Unavoidable...
More Detail

URLs, Names, Preferences, Session
Links in a Portlet

- Portlet can't generate “normal” URLs to itself
  - Normal links to external pages are OK
    - Leave the Portal or target new window

- Links to a Portlet are special
  - Links to content
  - FORM actions
    - “Targeted” to the Portlet
      - Portal still has to render other Portlets
      - Portal might need to mangle the URL

- Two types of URLs
  - Action – like processing FORM submit – runs processAction()
  - Render – displays content
Portlet URLs

- **Action URL**
  - `renderResponse.createActionURL()`
  - Causes the targeted Portlet's `processAction` method to be run before rendering any Portlets

- **Render URL**
  - `renderResponse.createRenderURL()`
  - Causes the Portlet to be rendered
  - Does not invoke `processAction`

- Can pass parameters, change window state, change modes...
  - Parameters only go to the targeted Portlet
Namespace Encoding

- Portlets generate HTML markup that must live on the same page with other HTML markup
- Requires names and identifiers to be name-spaced
  - So they are unique to the Portlet
  - Avoid collisions
  - HTML id's
  - Javascript variables and functions
- Prefix with `getNamespace()`
  - `renderResponse.getNamespace() + "ident"`
Forms in a Portlet

Forms should use POST

➢ Don't use GET method
  • Portal may encode state information in query string
➢ Use POST with ActionURL and parameters
  • Encode any ids and javascript

```java
PortletURL url = portletResponse.createActionURL();
url.addParameter("searchUsing", "google");
url.setWindowState( WindowState.MAXIMIZED );
url.setPortletMode( PortletMode.VIEW );
writer.print( "<FORM METHOD="POST" ACTION="" + url.toString() + "" >" );
writer.print( "<INPUT TYPE="TEXT" NAME="keywords">" );
writer.print( "</FORM>" );
```
Portlet Preferences

- Configuration for a Portlet instance
  - Persistent customization of a Portlet's view or behavior
  - Declared in deployment descriptor (portlet.xml)
    - Specific to Portlet instance
    - Allows deploying multiple instances of the same Portlet with different preferences
    - Example:
      - News Feed Portlet
      - News Stream RSS URL as a Preference
      - Deployed 3 times – for Slashdot, BBC, and TheServerSide

- NOTE Preferences are Configuration Data
  - NOT a substitute for a Database
Portlet User Preferences

- Preferences are also scoped to the User
  - For example, in News Feed Portlet
    - EDIT page could allow user to change
      - The RSS URL
      - Max number of news items

- No distinction made in API to determine:
  - Portlet-container supplied Preference defaults
  - Preferences from deployment descriptor
  - Preferences supplied by Portal
  - User-specific Preferences
Using Preferences

- Retrieve PortletPreferences object from PortletRequest
  ```java
  PortletPreferences prefs = portletRequest.getPreferences();
  ```

- Get values
  ```java
  String[] stocks = prefs.getValues("stockSymbols",
  new String[] {"BEAS", "SUNW", "IBM" });
  ```

- Get attribute names or Map of name/value pairs
  ```java
  Enumeration getNames()
  Map getMap()
  ```

- Can use Preferences in any Portlet method, Mode, State
Preferences are String Arrays

String[] getValues(String name, String[] default)
void setValues(String name, String[] vals)

- It is up to the developer (you) to ensure:
  - Manage multi- vs. single-valued data
  - Conversions to appropriate types (int, date, etc.)
    - PreferencesValidator can help

- Convenience getValue / setValue methods for single-valued data
  - getValue() returns getValues()[0]
  - setValue(val) is setValues( new String[] { val } )
Modifying Preferences

Preferences may be modified only during `processAction()` method

Methods:

- `boolean isReadOnly(String name)`
- `void setValue(String name, String val)`
- `void setValues(String name, String[] vals)`
- `reset(String name)`

- Restore attribute to default value
  - Up to the vendor to determine the default
  - Probably in deployment descriptor
  - Delete the attribute if no default available
Persisting Preference Changes

- Nothing is persisted until you call store()
  - Atomic persistence of all changes
  - May throw exception if store is not possible
    - For example, no user is logged in
    - No good way to pre-determine chance of success
    - No consistent way to determine reason for failure
- If store() is not called, changes are discarded on return from processAction()
HTTP Session Attributes

Portlets can access the HTTP Session

- PortletSession is mirror of HttpSession
- Uses one of two scope identifiers
  - PORTLET_SCOPE
    - This is the default
    - Attributes names are name-spaced to the individual Portlet instance
      - javax.portlet.p.<ID>?<ATTRIBUTE_NAME>
      - PortletSessionUtils for access from a Servlet
  - APPLICATION_SCOPE
    - Attributes in the normal HTTP session without name mangling
    - Can communicate between Portlets or Servlets
    - Should only change Session from processAction()
Code Break

Picture Portlet

➢ Display picture specified by Preference
➢ EDIT mode to change preference
  • FORM
Portlets, Servlets, and JSPs

Dispatching Requests
Servlets and JSPs

- Portlet can include Servlets, JSPs, other pages
- Allows Portlet to
  - Externalize presentation markup
  - Reuse existing JSP pages and Servlets
- Restrictions
  - Must output markup fragments
  - No access to some HTTP Request/Response data
    - Protocol, RemoteAddr, RemoteHost, RealPath, RequestURL
    - CharacterEncoding, ContentType, ContentLength, etc.
    - InputStream, Reader
    - Cookies, Response Header, etc.
PortletRequestDispatcher

From render() method:

```java
String path = " /cart.jsp?orderid=" + orderId;

PortletRequestDispatcher rd = context.getRequestDispatcher(path);

rd.include(renderRequest, renderResponse);
```
Included Servlets and JSPs

- Can access Portlet objects
  - Request Attributes set by PortletRequestDispatcher
    - `javax.portlet.config` – the PortletConfig
    - `javax.portlet.request` – the RenderRequest
    - `javax.portlet.response` – the RenderResponse

- Share Session data
  - `APPLICATION_SCOPE`
  - Use PortletSessionUtils to encode PORTLET_SCOPE attribute names
Portlet Tags for Included JSPs

- `<portlet:defineObjects/>`
  - Creates variables for
    - renderRequest
    - renderResponse
    - portletConfig

- `<portlet:actionURL ... />`
  - `<portlet:renderURL .../>`
  - Creates action and render URLs
  - `<portlet:param .../>` sub-tag for request parameters

- `<portlet:namespace/>`
  - For namespaces in identifiers in Forms, JavaScript, etc.
JSP Portlet Pattern

- Portlet class contains `performAction()`
  - Process forms, other actions
  - Modify Preferences and Session data
  - Maintain state for multi-page forms

- Render method dispatch to JSPs
  - For presentation markup
  - But JSP probably not generally reusable
    - JSP must only generate fragments
    - JSP probably strongly tied to a Portlet or set of Portlets

- JSP is hosted by WebApp, accessible from outside
  - Rely on unpublished URLs to hide it
Code Break

- Picture Portlet with rendering externalized to JSP
  - Separate JSP page for VIEW and EDIT modes
  - SimpleJspPortlet base class does render dispatch
Other Topics

Content Types, Resource Bundles, Security, WSRP, etc.

... nearing the end.
Content Type

- Content Type is the type of data served by a request
  - text/html, text/plain, text/vnd.wap.wml, etc.
- Unlike a Servlet or JSP, a Portlet does not get to decide its own content type
  - Decided by the Portal which contains it
- Must declare which Content Type(s) it supports
- Must query portal for which types are allowed for a particular request
- Portlet should render content of that type
Internationalization / Localization

- Portlet can declare a ResourceBundle
  - In Deployment Descriptor (portlet.xml)
- ResourceBundle is used to override portlet-info elements in Deployment Descriptor
  - title, short-title, keywords
  - Used in Portlet's title bar and for searching, etc.
- Portlet can also access resources from the bundle
  - Using PortletConfig.getResourceBundle() 
- Preferences can also be localized in the ResourceBundle
Portlet Security

- Programmatic security APIs like Servlet
  - getRemoteUser, getUserPrincipal,.isUserInRole
  - Role mapping in portlet.xml descriptor

- Declarative security for Transport (SSL)
  - NONE, INTEGRAL, CONFIDENTIAL
  - No definition of how Portal handles INTEGRAL or CONFIDENTIAL Portlet with http: request

- No Declarative Security for Portlet Authorization
  - Portlet needs to decide how it acts
    - Display Nothing? Default Content? "Please Login"?
    - Error? Login dialog? Something Else?
Portlet Caching

- Portlet Container may provide Caching for rendered content fragments
  - It is optional
  - Cache is per-Portlet and per-User

- Portlets declare that they want caching
  - Set expiration-cache element in portlet.xml
    - Time in seconds
    - Can override cache time in RenderResponse

- Cache is discarded when a Portlet URL targets the Portlet explicitly
  - actionURL or renderURL
WSRP

- Web Services for Remote Portlets
- OASIS spec
- Related to, but not part of, JSR-168
- Allow Portal to aggregate Portlets from other servers
Summary

- Portlet Spec closely related to Servlet
  - Portal App runs in a WebApp
  - Many similar concepts
  - New Programming model
    - Since Portlet must coexist with other Portlets
      - Names, URLs, Actions, etc.

- Portlets generate Fragments, not Pages
- Standard Portlets should be portable across Portal vendors
Web References

- Portlet Spec

- Pluto
  - Portlet Reference Implementation
  - http://jakarta.apache.org/pluto

- Servlets
  - http://java.sun.com/products/servlet

- WSRP
The End – Thank You

- Please fill out evaluations

- Example Code
  - On the conference CDROM
  - http://www.avitek.com/landers

- dave.landers@4dv.net
- dave.landers@bea.com