

## JSR-286: Portlet Specification 2.0

Upcoming enhancements and new features  
for Portal and Portlet Developers

Ate Douma

JSR-286 Expert Group Member

Software Architect

Hippo - Open Source Content Management Software

[ate@apache.org](mailto:ate@apache.org) / [a.douma@hippo.nl](mailto:a.douma@hippo.nl)





## Preamble

*This presentation only represents the current thinking of the JSR-286 Expert Group. The final specification may differ from the content of this presentation.*

## Agenda

- A short summary of JSR-168: Portlet Specification 1.0
- What is missing from JSR-168
- JSR-286: Enhancements and new features for 2.0
- Questions and Answers



## JSR-168 - A short summary

### PLT.2.1 What is a Portal?

*“ A portal is a web based application that -commonly- provides personalization, single sign on, content aggregation from different sources and hosts the presentation layer of Information Systems.”*



## JSR-168 - A short summary

### PLT.2.2 What is a Portlet?

*“A portlet is a Java technology based web component managed by a portlet container that processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems.”*



## JSR-168 - A short summary

### PLT.2.3 What is a Portlet Container

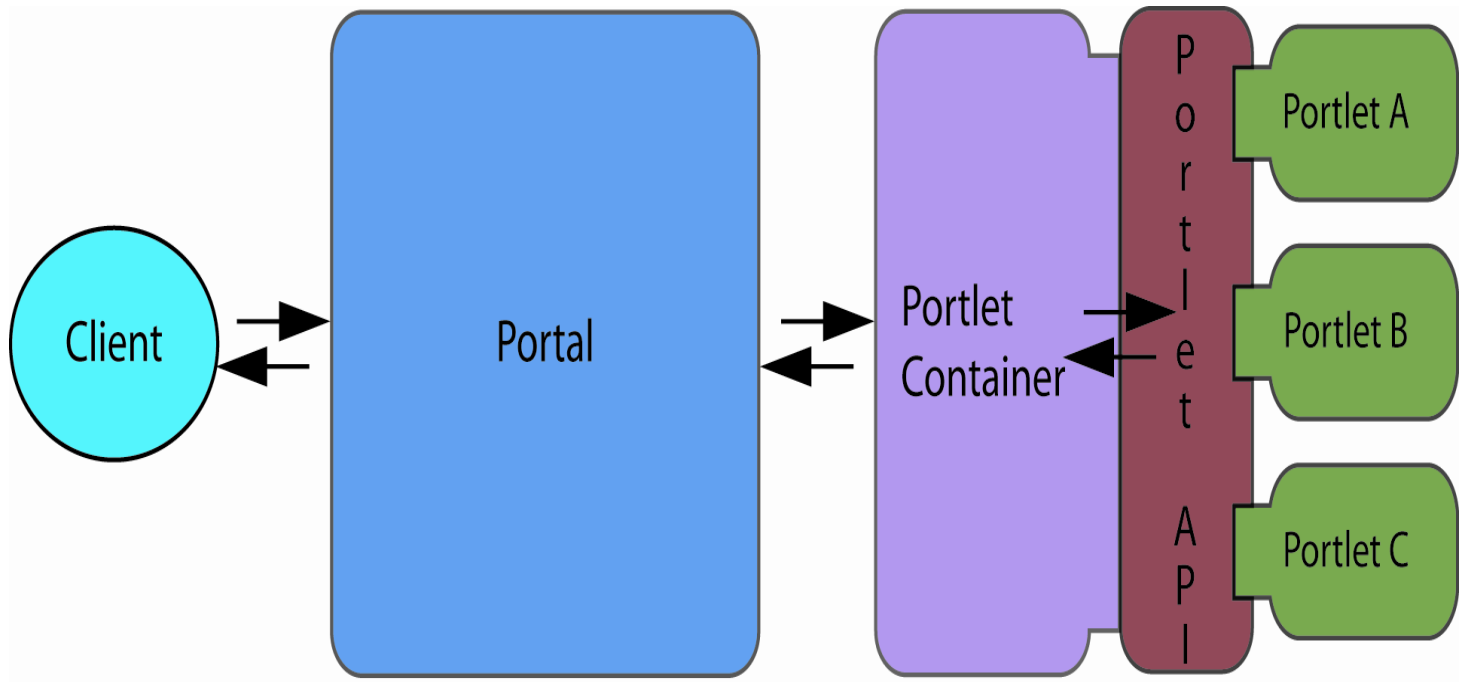
*“A portlet container runs portlets and provides them with the required runtime environment. A portlet container contains portlets and manages their lifecycle.”*

...

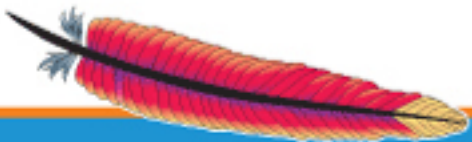
*“A portlet container receives requests from the portal to execute requests on the portlets hosted by it.”*



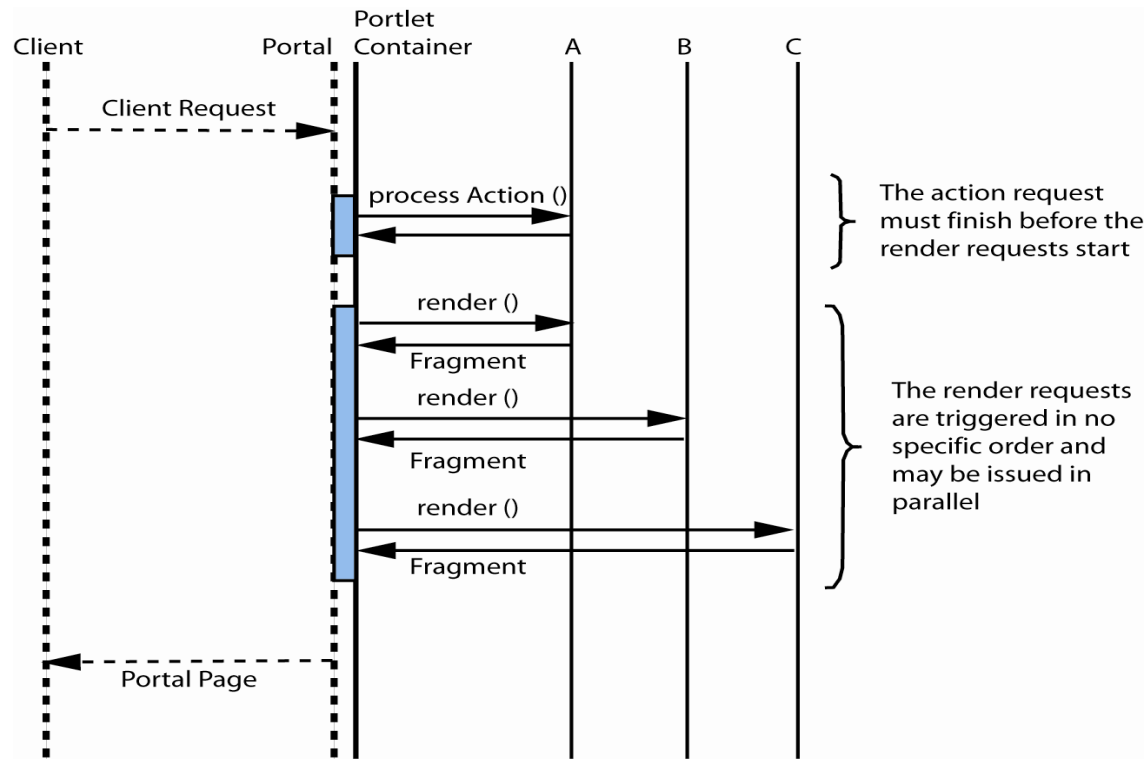
## JSR-168 - A short summary



*(Portlets and Apache Portals, Manning 2005)*



## JSR-168 - A short summary



*Request processing*

*(Portlets and Apache Portals, Manning 2005)*





## JSR-168 - What is missing?

- inter-portlet communication / portlet coordination
- serving non-html resources (pdf, doc, images etc.)
- portlet filters
- contributing javascript or css to <head>, using cookies
- proper support for common web frameworks
- AJAX



## JSR-168 - What is missing?

### Inter-portlet communication

- only supported within the same portlet application using session attributes
- target portlets will only “see” messages during next render request
- portlets cannot (should not) update their state during a render request: “event” handling not really possible



## JSR-168 - What is missing?

### Serving non-html resources

- A portlet can only render html fragments
- Have to fallback/delegate to the servlet container
- Requires coordination between portlet and servlet



## JSR-168 - What is missing?

Contributing to <head>, setting cookies

- javascript or css can only be embedded within the content markup; no body onLoad handling hooks
- API forbids adding cookies: only client side setting of cookies using javascript is possible



## JSR-168 - What is missing?

Proper support for common web frameworks

- Most web frameworks are still primarily (or only) Servlet API oriented, and with reason
- Servlet dispatching not supported from processAction
- Needs Portals Bridges or similar solutions
- Split between Action and Render makes migration painful
- “Unaware” servlets only “see” the application scope session
- JSP/JSTL support very limited:

```
<c:out value="<%= ((FooBean)renderRequest.getSession()  
                .getAttribute("fooBean",PortletSession.PORTLET_SCOPE))  
                .getBeanValue() %>"/>
```



## JSR-168 - What is missing?

### AJAX

- Many portals actually **do** support AJAX now
- But: none of that is support through the spec.
- Requires custom / portal specific extensions
- No portable solution possible



## JSR-286

- Portlet Specification 2.0
- Expert Group started January 2006
- 1<sup>st</sup> Early Public Draft was released August 2006 covering most, but not yet all, planned features
- Several public but not-official public early drafts
- Currently at revision 14, d.d. 2007-04-27, now covering (almost) all planned features
- Final Release planned (late) Summer 2007
- RI done under Apache Pluto umbrella with help from a developers group at University of Jena



## JSR-286: Main Features

- Alignment with J2EE 1.4, WSRP 2.0
- Portlet coordination
  - Public Render Parameters
  - Portlet Events
- Resource serving (which can be used for partial AJAX support)
- Portlet Filters
- Extended Cache support
- Container runtime options
- Improved support for common web frameworks
- ... and more ...





## JSR-286: Public render Parameters

- Coordination of render state with other portlets across web applications
- Limited to String values
- Defined in portlet.xml using QNames:

```
<public-render-parameter>
  <identifier>foo</identifier>
  <name xmlns:x="http://acme.com/parameters">x:foo.2</name>
</public-render-parameter>

<portlet>
  <portlet-name>Portlet-A</portlet-name>
  ...
  <supported-public-render-parameter>
    <name>foo</name>
  </supported-public-render-parameter>
</portlet>
```



## JSR-286: Portlet Events

- A portlet declares the events it wants to receive or emit in portlet.xml using QNames:

```

<event-definition>
  <name xmlns:x="http://acme.com/events">x:foo</name>
  <java-class>java.lang.String</java-class>
</event-definition>
<portlet>
  <portlet-name>Portlet-A</portlet-name>
  ...
  <supported-processing-event>
    <name xmlns:x="http://acme.com/events">x:foo</name>
  </supported-processing-event>
  <supported-publishing-event>
    <name xmlns:x="http://acme.com/events">x:foo</name>
  </supported-publishing-event>
</portlet>

```

- Emitting dynamic non-declared events is allowed too



## JSR-286: Portlet Events (cont.)

- Allows wiring of portlets at runtime
- The portal / portlet container will act as broker
- The portal or portlet container can also emit events
- Formally not 100% reliable, i.e. no guarantee of delivery (mainly for WSRP, “local” portals are less limited)
- New 3<sup>rd</sup> lifecycle phase: before rendering
- API:

```
EventPortlet.processEvent(EventRequest req,EventResponse res)  
StateAwareResponse.setEvent(String name,Object value)  
StateAwareResponse.setEvents(Map events)
```



## JSR-286: Portlet Events (cont.)

- Can be send from both `processAction` and `processEvent`
- State changes are allowed during `processEvent`
- Will be delivered in “order” as send out by the Portlet



## JSR-286: Resource Serving

- New type of PortletURL and request handler
- API:

```
PortletResponse.createResourceURL()  
ResourceServingPortlet.serveResource(ResourceRequest, ResourceResponse)
```

- Extends the render phase, NOT a new life-cycle phase
- POST, PUT, DELETE supported
- Should not change shared portlet state
- Additional URL parameters are specific for the request
- Full control over request and response headers
- Can be used for binary data or “readonly” AJAX (cannot change navigational state)



## JSR-286: ~~Native AJAX~~ support

Was intended and discussed at large

- But: will not be part of this spec
  - AJAX request identification
  - synchronizing state changes to the client for other portlets on the page
  - XMLPortletRequest or plain XMLHttpRequest
  - integration with popular AJAX toolkits
  - Too many not yet known or fully thought through use-cases
- Instead: guidelines will be provided for common use-cases with possible solutions (in concerto with WSRP 2.0 team)
- Let the community determine a feasible API first



## JSR-286: Portlet Filters

- Allow on the fly transformations of information in both the request to and the response from the portlet
- Modelled after Servlet filters
- Defined in portlet.xml:

```
<filter>
  <filter-name>Event Filter</filter-name>
  <filter-class>com.acme.EventFilter</filter-class>
  <lifecycle>EVENT_PHASE</lifecycle>
</filter>

<filter-mapping>
  <filter-name>Event Filter</filter-name>
  <portlet-name>SamplePortlet</portlet-name>
</filter-mapping>

<filter-mapping>
  <filter-name>Event Filter</filter-name>
  <portlet-name>OtherPortlet*</portlet-name>
</filter-mapping>
```



## JSR-286: Extended Cache support

- Allow public cached content for multiple users

```
<portlet>
  ...
  <expiration-cache>
    <expiration-time>300</expiration-time>
    <scope>public</scope>
  </expiration-cache>
</portlet>
```

- Support validation based caching using ETAG
- API:

```
response.getCacheControl()
  .getExpirationTime(), .setExpirationTime()
  .getPublicScope(), .setPublicScope(boolean)
  .getETag(), .setETag(String)
  .useCachedContent(), .setUseCachedContent(boolean)
```





## JSR-286: Better web framework support

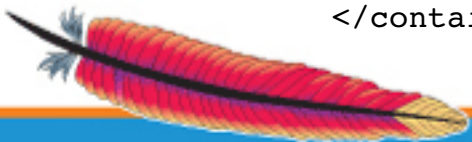
- Allow servlet dispatching during all lifecycle calls:
  - `processAction`
  - `processEvent`
  - `render`
  - `serveResource` (including using `forward`)
- Optional providing a `PORTLET_SCOPED` session to servlets (*as container runtime option, see next slide*)
- Extended JSP tag library
  - `<portlet:defineObjects/>`
  - `<portlet:action|renderURL copyCurrentRenderParameters="true" escapeXml="false">`
  - `<portlet:resourceURL>`



## JSR-286: Container runtime options

- Defineable in portlet.xml on both application and portlet level
- Queryable from PortletContext
- Three predefined options:
  - `javax.portlet.escapeXml` (for backwards compatibility)
  - `javax.portlet.renderHeaders`
  - `javax.portlet.includedPortletSessionScope`

```
<container-runtime-option>  
  <name>javax.portlet.includedPortletSessionScope</name>  
  <value>PORTLET_SCOPE</value>  
</container-runtime-option>
```



## JSR-286: ... and more ...

- Create PortletURLs within all life-cycle phases
- `PortletRequest.getWindowID()`
- `RenderResponse.setNextAvailablePortletModes()`
- Contributing to HTTP and HTML headers: `doHeaders(...)`
- Java 5 annotation support for events in `GenericPortlet`:  

```
@ProcessEvent(name="{http://acme.com/events}foo")  
  
void processFoo(EventRequest request, EventResponse response)  
    throws PortletException, java.io.IOException;
```
- JSR-188: Composite Capability/Preference Profiles:  
request attribute `PortletRequest.CCPP_PROFILE`



## More information

- This presentation and more links  
<http://www.hippoportal.org/display/PORTAL/JSR-286>  
  
JSR-286: Portlet Specification 2.0  
<http://jcp.org/en/jsr/detail?id=286>
- Online current early draft (non-official)  
<http://ipc658.inf-swt.uni-jena.de/spec/>
- Apache Pluto  
<http://portals.apache.org/pluto/>
- RI development  
<http://svn.apache.org/repos/asf/portals/pluto/branches/1.1-286-COMPATIBILITY/>



# ApacheCon



## Questions and Answers

EU 2007

JSR-286: Portlet Specification 2.0