Supporting JSF 2.x

Issues for adding JSF 2.x support to the Bridge

1. Distinguish targets of Ajax requests:
   As a general rule, the Bridge#doFacesRequest(ResourceRequest, ResourceResponse) method needs to determine if the ResourceRequest is for Ajax, or a JSF 2 resource.
   If it's for a JSF 2 resource, then the ResourceHandler#handleResourceRequest() method should be called. Otherwise, the Faces lifecycle should be executed, and also render-response.
   For more info, see: http://svn.portletfaces.org/svn/portletfaces/bridge/portletfaces-bridge-impl/trunk/src/main/java/org/portletfaces/bridge/BridgeImpl.java
   No changes to the Public API are necessary. Only the Spec document would have to discuss this.
2. File upload support in JSF 2.2:
   Just wanted to mention that JSF 2.2 includes support for full postback multipart/form-data file upload, and also Ajax-based file upload.
   The full postback design is based on work I did in my bridge, and contributed to Mojarra. Ed Burns took out the dependency on commons-fileupload in favor of making JSF 2.2 require Servlet 3.0 file upload capabilities.
   I don't think the Ajax file upload feature is done yet, so I don't have any info on how that might impact the bridge, if at all.
3. Properly Handle Resources: Custom ResourceHandler?
   I remember this as was one of the more challenging issues to deal with. The main problem is that there is an assumption in the JSF 2 Spec where Faces thinks that JSF 2 resource URLs will always have a recognizable pattern, like this:
   http://hostname/webappcontext/javax.faces.resource?ln=libraryname&v=1234
   The problem is that in a portlet environment, the portlet container is in full control of URLs generation, and "/javax.faces.resource" isn't necessarily going to be there.
   So I worked around this problem by having a custom ResourceHandlerImpl in the chain-of-responsibility. Also I had to modify the behavior of ExternalContext#encodeResourceURL(String) to add "javax.faces.resource" as a URL parameter, rather than a URL pattern.
   Back in March of 2009, I asked Ed Burns about issues related to this, and he wrote:
   > "We thought of this and imagined that the Portlet would provide a custom > ResourceHandler. That would be the entry point to control how the > resource requests are appropriately directed."
   So the portlet bridge is going to have to handle serving up JSF 2 resources in some manner. Just wanted to let you know the way I did it.

Improvements JSF 2.x can make to facilitate Bridge implementation

1. Provide extensible/overridable core Flash implementation for Bridge to base itself on.

The main problem is that in JSF 2, instances of the Flash scope are retrieved from the following method:
http://javaserverfaces.java.net/nonav/docs/2.0/javadocs/javax/faces/context/ExternalContext.html#getFlash()
This means that the bridge is forced to provide its own implementation of the Flash scope, because there's no way for the bridge to get at the Flash implementation from Mojarra or MyFaces.
I started down the road of creating my own bridge Flash implementation, but the internals are somewhat complicated. So instead, I worked around the issue by creating instances of the Mojarra and MyFaces Flash scope implementations using reflection. It worked fine, so I think creating a FlashFactory in the JSF 2.2 API is the way to go. That's what I plan on suggesting to Ed.