



Moving JavaServer Faces Forward

JSF 2.2 BOF





Latin America 2011

December 6–8, 2011

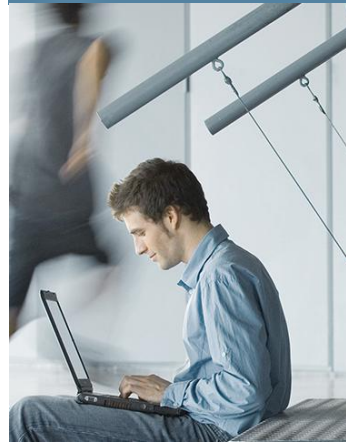
Tokyo 2012

April 4–6, 2012

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

- Where is JSF in its lifecycle?
- What was in JSF 2.1?
- What will be in JSF 2.2?
 - What's done so far?
 - What's up next?
- Q&A



Where is JSF in its lifecycle?

Mature and stable

- JSF 2.0 was the last BIG FEATURE release for JSF
- From now on, JSF will be maintained using smaller, more frequent releases.
 - JSF 2.1, released Q4 2010, JCP Minor Revision
 - JSF 2.2, slated for release in 2012, full JSR
 - Requires JavaEE 6, will release before JavaEE 7
 - JSF 2.3, will target JavaEE 7

What was in JSF 2.1?

Minor Revision <http://javaserverfaces.java.net/nonav/docs/2.1/javadocs/>

- Allow replacing the Facelet cache
 - `<facelet-cache-factory>`, `FaceletCacheFactory`
- Allow treating JSP Document files (.jspx) as Facelet files
 - `<facelets-processing>`
- JSF Managed Beans must be CDI injectable
- Specify that EL in resources must be evaluated
- `TransientStateHelper/TransientStateHolder`
- `UIData` transient state saving

What will be in JSF 2.2?

What's already in?

- Started on 14 March 2011
- Early Draft Review 8 November 2011
- Small changes in 2.2
 - `ExternalContext.getApplicationContextPath()`
 - Allow composite components to be referenced in Facelets `.taglib.xml` files, allowing the mixing of composite and non-composite components in one library
 - Composite component metadata improvements

What will be in JSF 2.2?

What's already in?

- Bigger Changes
 - New tag `<f:viewAction />`.
 - Cross Site Request Forgery Protection
 - More artifacts are injectable via CDI
 - FaceletFactory now in the API
 - FaceletsResourceResolver annotation
- How to browse the latest spec
 - Visit <http://maven.java.net/>
 - Search for “javax.faces-api”
 - Browse to 2.2-SNAPSHOT javadoc artifact
 - Use the “Archive Browser” tab

What will be in JSF 2.2?

`<f:viewAction />`

- Directly taken from Seam's Faces Module
`<s:viewAction>`
- Think of it like a button that automatically clicks itself when the page loads

What will be in JSF 2.2?

Cross Site Request Forgery Protection

- What is Cross Site Request Forgery (CSRF, pronounced SEE-surf)?
 - http://en.wikipedia.org/wiki/Cross-site_request_forgery
 - Trick the browser into sending requests that the user did not actually intend to initiate

What will be in JSF 2.2?

Cross Site Request Forgery Protection

- How does JSF protect your app against this attack?
 1. It already does and always has!
POSTback is a virtue. JSF 2.2 just makes encryption of the view state on by default

What will be in JSF 2.2?

Cross Site Request Forgery Protection

- How does JSF protect your app against this attack?
 2. For non-POSTback requests, the <protected-views> element comes to the rescue.
 - ViewHandler asks “Is this view protected?” if so, proceed.
 - If not, inspect the Referer[sic] header. If coming from a protected view, proceed.
 - If not, does the referrer at least originate in this web app? If so, proceed.
 - If no Referer header, look for `ResponseStateManager.NON_POSTBACK_VIEW_TOKEN_PARAM`
This query parameter will be present because of these changes to `ResponseStateManager.getState()`

What will be in JSF 2.2?

More Artifacts Injectable via CDI

TABLE 5-3 JSF Artifacts Eligible for Injection

Artifact Type	Injected if declared in application configuration resources (faces-config.xml)	Injected if declared via Annotation
<code>javax.faces.event.ActionListener</code>	yes	Not Applicable
<code>javax.el.ELResolver</code>	yes	Not Applicable
<code>javax.faces.application.NavigationHandler</code>	yes	Not Applicable
<code>javax.faces.application.ResourceHandler</code>	yes	Not Applicable
<code>javax.faces.application.StateManager</code>	yes	Not Applicable
<code>javax.faces.event.SystemEventListener</code>	yes	Not Applicable
<code>javax.faces.component.UIComponent</code>	yes	yes
<code>javax.faces.component.behavior.Behavior</code>	yes	yes
<code>javax.faces.convert.Converter</code>	yes	yes
<code>javax.faces.validator.Validator</code>	yes	yes

What will be in JSF 2.2?

FaceletFactory now in the API

- Yes, it is now programmatically possible to instantiate a Facelet and call `apply()` on it.
- Why would you need to do this?
- Because the result of `Application.createComponent()` is not always enough
 - No automatic ajaxification
 - No child handlers
 - Other cases



What's up next in JSF 2.2?



What's up next in JSF 2.2?

Sketched but not fully specified

- Task Flows
- File Upload
- Multi Templating

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Q&A