JBoss WORLD
CHICAGO 2009

FOLLOW US:

TWITTER.COM/REDHATSUMMIT

TWEET ABOUT US:

ADD #SUMMIT AND/OR #JBOSSWORLD TO THE END OF YOUR EVENT-RELATED TWEET

presented by

JBoss WORLD
CHICAGO 2009

THE BEST KEPT SECRETS OF SEAM, RICHFACES, JSF AND FACLETS

Dan Allen
Sr. Software Engineer, Red Hat

Jay Balunas
RichFaces project lead, Red Hat

Sept 2, 2009

presented by

Displaying the Seam version

Implementation-Version property in Seam JAR manifest

Seam.class.getPackage().getImplementationVersion()

Stored in application-scoped variable at startup

Variable name: org.jboss.seam.version

Scope: application

EL: #{org.jboss.seam.version}

TIP
Use the same approach to get the implementation version, specification version, and even vendor name from other libraries.

Custom EL functions (1)

Facelets has mechanism for registering EL functions

```
#{str:encodeURL(url)}
```

JBoss EL supports...

- parametrized methods
- static methods

Using Seam component is simpler

- @BypassInterceptors
- Static methods

Custom EL functions (2)

Define a stateless Seam component

```
@Name("stringutils")
@Scope(ScopeType.STATELESS)
public class StringUtils {
   public static void truncate(String s, int max) {
      ...
   }
}
```

Invoke method using literal syntax

```
#{stringutils.truncate(name, 25)}
```

Reuse existing libraries

```
<component name="stringutils"
   class="org.apache.commons.lang.StringUtils"
   scope="stateless"/>
```

Output context path

java.net Facelets

Required to reference web asset from HTML tag

The “JSF way” is to read from ExternalContext

```
#{facesContext.externalContext.requestContextPath}
```

Facelets provides a shorthand

```
#{request.contextPath}
```

```
<link rel="shortcut icon"
     href="#{request.contextPath}/favicon.ico"/>
```

NOTE
Facelets supports the same implicit objects as JSP, plus the objects introduced by JSF.

Set response content type

java.net Facelets

Facelets allows content type to be set on `<f:view>`

Makes serving XML with Facelets is easy!

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://purl.org/atom/ns#" version="0.3"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:f="http://java.sun.com/jsf/core">
  <f:view contentType="application/atom+xml">
    <title>News Feed</title>
    <ui:repeat var="_entry" value="#{entries}">
    <entry>
        <title>#{_entry.title}</title>
        <summary
  type="text/plain">#{_entry.body}</summary>
        ...
    </entry>
    </ui:repeat>
  </f:view>
</feed>
```

Spacer for Facelets

java.net Facelets

Facelets is horrible about eating whitespace

```
<h:commandLink value="Link 1"/>
<h:commandLink value="Link 2"/>
```

→ `Link 1Link 2`

Solution: Use EL expression that inserts a space

```
<h:commandLink value="Link 1"/>
#{' '}
<h:commandLink value="Link 2"/>
```

→ `Link 1 Link 2`

Shorthand: Bind space to single-character variable

```
<factory name="_" value=" " scope="APPLICATION"/>

#{' '} == #{_}
```

NOTE
  is not appropriate because it inserts too much space or prevents a line break.

Common Facelets JAR (1)

java.net Facelets

Leverage pluggable resource resolver to locate template

```
public class ClasspathResourceResolver extends
   DefaultResourceResolver implements ResourceResolver {

   public URL resolveUrl(String resource) {
      URL resourceUrl = super.resolveUrl(resource);
      if (resourceUrl == null) {
         resource = resource.startsWith("/") ?
            resource.substring(1) : resource;
         resourceUrl = Thread.currentThread()
            .getContextClassLoader()
            .getResource("META-INF/" + resource);
      }
      return resourceUrl;
   }

}
```

Common Facelets JAR (2)

java.net Facelets

Register the resolver using a context param in web.xml

```
<context-param>
   <param-name>facelets.RESOURCE_RESOLVER</param-name>
   <param-value>ClasspathResourceResolver</param-value>
</context-param>
```

Package templates in a JAR file and reference them

```
<ui:composition template="/templateInJAR.xhtml">
...
</ui:composition>
```

NOTE
Facelets automatically loads tags mapped to source templates that are in the META-INF directory of a classpath entry.

Iterating a java.util.Set (1)

JSF standardized UIData on List instead of Collection

- Leaves Set out in the cold
- Many ORM associations are sets

How do you iterate a Set using a UIData component?

- Seam
- JBoss EL

Iterating a java.util.Set (2) – Outjection

@DataModel outjection wraps Set in SetDataModel

- Must be triggered by a factory or action invocation

```
@Name("employeeList")
public class EmployeeList {
   @DataModel Set employees;

   public void search() {
      employees = ...;
   }
}
```

Does not address Set association (e.g. employee jobs)

Iterating a java.util.Set (3) – Component

Define SetDataModel as component to wrap collection

```
<component name="jobs" scope="event"
   class="org.jboss.seam.jsf.SetDataModel">
   <property
  name="wrappedData">#{employee.jobs}</property>
</component>
```

Only works well for global data

Iterating a java.util.Set (4) – Factory

Uses dataModels component to wrap collection

```
<factory name="jobs" scope="event"
   value="#{dataModels.getDataModel(employee.jobs)}"/>
```

Again, only works well for global data

Iterating a java.util.Set (5) – EL Projection

Convert any Collection to a List using an EL projection

```
<h:dataTable var="_job" value="#{employee.jobs.{j|j}}">
   <h:column>
      #{_job.title}
   </h:column>
</h:dataTable>
```

Best solution for traversing object graphs

Printing the iteration index (1)

UIData component doesn’t have iteration variable

Solution #1: Use a component reference

```
<h:dataTable id="lineItems" var="lineItem"
   value="#{orderBean.lineItems}">
   <h:column>
      Row: #{uiComponent['lineItems'].rowIndex + 1}
   </h:column>
   ...
</h:dataTable>
```

Printing the iteration index (2)

UIData component doesn't have iteration variable

Solution #2: Use RichFaces row key iteration variable

```
<rich:dataTable id="lineItems" var="lineItem"
  rowKeyVar="i"
   value="#{orderBean.lineItems}">
   <h:column>
      Row: #{i + 1}
   </h:column>
   ...
</rich:dataTable>
```

Available on any RichFaces UIData component

- aj4:repeat
- rich:dataList
- ...etc

Printing the iteration index (3)

UIData component doesn't have iteration variable

Solution #3: Use varStatus on `<ui:repeat>` (JSF 2.0)

```
<table>
   <ui:repeat id="lineItems" var="lineItem" varStatus="it"
      value="#{orderBean.lineItems}">
      <tr>
         <td>Row: #{it.index}</td>
         ...
      </tr>
   </ui:repeat>
</table>
```

Provides complete iteration status

first, last, begin, end, step, current, index and count

Must abandon data table semantics

Recover from expired view

JSF views can become “stale”

Use Seam’s pages.xml exception handling to recover

```
<exception class="javax.faces.application.ViewExpiredException">
   <redirect view-id="#{org.jboss.seam.handledException.viewId}">
      <message severity="warn">
         Your session timed out. A new session
         has been created and you were redirected
         back to the requested page.
      </message>
   </redirect>
</exception>
```

Action on navigation

Goal: Invoke action when navigate away from page

Redirect to fictitious view ID with page action

Redirect from page action to real view ID

```
<page view-id="/register.xhtml">
   <navigation from-action="#{registration.register}">
      <rule if-outcome="success">
         <redirect view-id="/postRegister.xhtml"/>
      </rule>
   </navigation>
</page>
<page view-id="/postRegister.xhtml">
   <action execute="#{quotaManager.allocate}"/>
   <navigation from-action="#{quotaManager.allocate}">
      <redirect view-id="/accountHome.xhtml"/>
   </navigation>
</page>
```

Load seed data in development

Application-scoped, debug, startup component

Load data in @PostConstruct (or @Create) method

```
@Name("dataLoader")
@Scope(ScopeType.APPLICATION)
@Install(debug = true)
public class DataLoader {
    @In EntityManager entityManager;

    @Transactional @PostConstruct
    public void execute() {
        entityManager.persist(new Widget(...));
        ...
    }
}
```

Activated using <core:init debug="true"/>

Constructor double take

"Why is the constructor of my component called twice?"

Answer: Side-effect of Javassist object proxying

Solution: Put initialization code in a @PostConstruct (or @Create) method

Interceptors on sibling methods

Interceptors don’t “see” method calls invoked on **this**

Unwrapped instance shadows component’s name

Look up in explicit scope to get proxied instance

```
@Name("paymentProcessor")
public class PaymentProcessor {

   public void purchase() {
      PaymentProcessor thisComponent =
         (PaymentProcessor) Component.getInstance(
            "paymentProcessor", ScopeType.EVENT);
      thisComponent.schedulePayment();
   }

   @Asynchronous
   public void schedulePayment() {
      ...
   }
}
```

@Out and @DataModelSelection don't mix!

@DataModelSelection annotation does not inject null

Keep @Out and @DataModelSelection separate

```
@Name("courseDirectory")
@Scope(ScopeType.CONVERSATION)
public class CourseDirectory {
   @DataModel List<Course> courses;
   @DataModelSelection Course selectedCourse;
   @Out(required = "false") Course activeCourse;
   ...
   public void select() {
       activeCourse = selectedCourse;
   }
}
```

NOTE
Bijection may be triggered in other phases.

Interpolate the value of a message key

Substitute EL and/or positional parameters

```
NotInRange=#{user.name}, select a value between {0} and
  {1}
```

Step 1: Resolve message from resource bundle

```
String template = ResourceBundle.instance()
   .getString("NotInRange");
```

Step 2: Interpolate message template

```
String resolved = Interpolator.instance()
   .interpolate(template, 1, 1000);
```

Optionally use MessageFormat from Java API

```
String resolved = MessageFormat.format(template, 1, 1000);
```

Display constraint violation messages

Seam Hibernate Validator integrations

- JSF validation: `<s:validate>` and `<s:validateAll>`
- DML validation: persist, merge, update

Trap constraint violation and register JSF message

```
try {
   return super.persist();
}
catch (InvalidStateException ex) {
   for (InvalidValue iv : ex.getInvalidValues()) {
      FacesMessages.instance().add(iv.getMessage());
   }

   return "invalid";
}
```

Checking for a postback

JSF

ResponseStateManager has postback check (JSF 1.2)

```
FacesContext facesCtx = FacesContext.getCurrentInstance();
facesCtx.getRenderKit().getResponseStateManager()
   .isPostback(facesCtx)
```

As abbreviated using JBoss EL

```
#{facesContext.renderKit.responseStateManager.isPostback(
   facesContext)}
```

Actual logic performed

```
#{not empty param['javax.faces.ViewState']}
```

Convenience method added to FacesContext in JSF 2.0

```
FacesContext.getCurrentInstance().isPostback()
```

Checking for an Ajax request

Exposed via ajaxContext managed bean

#{ajaxContext.ajaxRequest}

See AjaxContext JavaDoc for additional methods

Component references using EL

Seam provides dynamic map to lookup component

```
#{uiComponent['lineItems']}
```

Equivalent to UIViewRoot#findComponent()

RichFaces offers same functionality with EL function

```
#{rich:findComponent('lineItems')}
```

Searches upwards to UIViewRoot, then back down

Additional RichFaces EL functions

rich:clientId – Return qualified clientId of component

rich:element – DOM element of component

rich:component – JavaScript control for component

Dynamic columns

Standard JSF data table only supports dynamic rows

Add components dynamically in Facelets build phase

```
<h:panelGrid columns="#{list.size}">
   <c:forEach var="row" value="#{list}">
      <rich:panel>#{row.name}</rich:panel>
   </c:forEach>
</h:panelGrid>
```

Use RichFaces to iterate columns in data table

```
<h:dataTable var="_restaurant" value="#{restaurants}">
    <rich:columns var="_criterion" value="#{criteria}">
        <f:facet name="header">#{_criterion}</f:facet>
        #{_restaurant[_criterion]}
    </rich:columns>
</h:dataTable>
```

Initially expand nodes in tree (1)

Use state advisor to send open hint (default is closed)

```
@Name("myTreeStateAdvisor")
public class MyTreeStateAdvisor implements TreeStateAdvisor {

    public Boolean adviseNodeOpened(UITree tree) {
       FacesContext ctx = FacesContext.getCurrentInstance();
       if (!ctx.getRenderKit()
             .getResponseStateManager().isPostback(ctx)) {
          TreeRowKey treeRowKey = (TreeRowKey)
  tree.getRowKey();
          if (treeRowKey == null || treeRowKey.depth() <= 2) {
             return Boolean.TRUE;
          }
       }

       return null;
    }

    ...
}
```

Optionally enforce a max expand depth.

Initially expand nodes in tree (2)

Bind the state advisor to the tree

```
<rich:tree switchType="ajax"
    ...
    value="#{treeBean.buildTree}"
    stateAdvisor="#{myTreeStateAdvisor}">
    ...
</rich:tree>
```

...or bind the method to the tree

```
adviseNodeOpened="#{myTreeStateAdvisor.adviseNodeOpened}"
```

Expand all descendant nodes (1)

Default behavior opens current node

Use data visitor to walk tree, recursively open nodes

Expand all descendant nodes (2)

Step 1: Define tree range to visit all nodes

```
public class CompleteTreeRange implements TreeRange {

   public boolean processChildren(TreeRowKey rowKey) {
      return true;
   }

   public boolean processNode(TreeRowKey rowKey) {
      return true;
   }
}
```

Expand all descendant nodes (3)

Step 2: Use data visitor to mark descendant nodes open

```
public class ExpandingDataVisitor implements DataVisitor {
   private TreeRowKey selectedKey;
   private UITree tree;

   public ExpandingDataVisitor(UITree tree, TreeRowKey key) {
      this.tree = tree;
      this.selectedKey = key;
   }

   public void process(FacesContext ctx, Object key, Object arg)
         throws IOException {
      TreeRowKey parentKey =
         (TreeRowKey) tree.getParentRowKey(key);

      if (selectedKey.isSubKey(parentKey)) {
         tree.queueNodeExpand(parentKey);
      }
   }
}
```

Expand all descendant nodes (4)

Step 3: Initiate tree walk from expand node listener

```
@Name("treeBean")
public class TreeBean {
   ...
   public void expandNode(NodeExpandedEvent e) {
      UITree tree = (UITree) e.getSource();
      ExpandingDataVisitor dataVisitor = new
  ExpandingDataVisitor(
         tree, (TreeRowKey) tree.getRowKey());
      try {
          tree.walk(FacesContext.getCurrentInstance(),
              dataVisitor, new CompleteTreeRange(), null, null);
      } catch (final IOException ioe) {
          ioe.printStackTrace();
      }
   }
}
```

Expand all descendant nodes (5)

Step 4: Bind the listener to the tree

```
<rich:tree switchType="ajax"
    ...
    value="#{treeBean.buildTree}"
    changeExpandListener="#{treeBean.expandNode}
    ...
</rich:tree>
```

Using RichFaces for style

RichFaces loads stylesheets on demand

Explicitly load stylesheet to style custom HTML

```
<a4j:loadStyle src="resource:///css/table.xcss"/>

<table class="rich-table">
...
</table>
```

Consult refdocs for complete list of stylesheets

Can also load JavaScript from JAR files

Partial page update via JavaScript function

Define JavaScript function that works like `<a4j:support>`

```
<a4j:jsFunction name="rate" reRender="rating">
   <a4j:actionparam name="stars"
      assignTo="#{ratingBean.stars}">
</a4j:jsFunction>
```

Invoke JavaScript function to update server state and UI

```
<button onclick="rate(1)" value="1 star"/>
<button onclick="rate(2)" value="2 stars"/>
<button onclick="rate(3)" value="3 stars"/>
<button onclick="rate(4)" value="4 stars"/>
<button onclick="rate(5)" value="5 stars"/>
```

Can also invoke JavaScript function after page update

- Function can accept serialized data from server

Invoking a component behavior

Components have client-side API (JavaScript)

```
<button onclick="rich:component('panel').show()"
   value="Show modal panel"/>
```

Can also attach behavior to another component event

```
<s:link id="link" value="Show modal panel"/>
<rich:componentControl attachTo="link" event="onclick"
   for="panel" operation="show"/>
```

More tips and FAQs

http://seamframework.org/Documentation/FAQs

http://seamframework.org/Documentation/KnowledgeBase

http://labs.jboss.com/community/wiki/RichFacesFAQ

http://www.jboss.org/community/wiki/RichFacesKnowledgeBase

JSF

http://wiki.apache.org/myfaces/FAQ

http://wiki.glassfish.java.net/Wiki.jsp?page=JavaServerFacesRI

Java java.net Facelets

http://wiki.java.net/bin/view/Projects/FaceletsFAQ

QUESTIONS?

TELL US WHAT YOU THINK:
REDHAT.COM/JBOSSWORLD-SURVEY