

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

Going Above and Beyond JSF 2 with RichFaces & Seam

Jay Balunas
Principal Software
Engineer

Lincoln Baxter, III
Senior Software
Engineer

JBoss, By Red Hat Inc

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Who's the big guy?



- Jay Balunas
 - RichFaces Project Lead
 - JBoss Core Developer
 - <http://in.relation.to/Bloggers/Jay>
 - <http://twitter.com/tech4j>
 - jbalunas@jboss.org



Co-author of
RichFaces
Dzone
RefCard

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Who's the tall guy?



Lincoln Baxter, III

([@lincolnthree](#), [blog](#))

- [Seam Faces](#) Lead
- Senior Software Engineer, JBoss
- JavaServer Faces EG Member
- Creator of [PrettyFaces](#) URL-rewriting extension for JSF & Java EE
- Co-founder of OcpSoft
 - <http://ocpsoft.com/>

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Questions...



Questions are welcome at anytime!*

* As long as:

“I’m not sure, but can find out”

“Lets discuss after the talk”

“I can point you in the right direction, but....”
are acceptable answers.

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





The Plan For Today

- Quick review of JavaServer Faces
- Review several features in more detail
 - For each feature
 - Where JSF 2.0 improved on JSF 1.X
 - Where the new feature may fall short
 - Show how RichFaces & Seam improve it
- Wrap up, and questions...

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Why?

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





What is JavaServer Faces (JSF)

- Component based UI framework
- Well defined request lifecycle
- Standards based (JCP)
- Many extension points
 - Component libraries
 - Phase listeners
 - View handlers
 - Renderers

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





What's so great about JSF 2.0

- Annotations instead of XML
 - Facelets integration
 - Composite components & templates
 - JSP deprecated
 - View parameters
 - Built in resource handling
 - Project stages
- Integrated AJAX support
 - New event systems
 - Bean validation
 - Exception handling
 - Navigation updates
 - Behaviors framework
 - EL access to component values

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- [An Introduction](#)
- Built in Ajax
- Built in Resource Handling
- Component Development
- Client Side Bean Validation
- Summary & Project Updates

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Lighting Review

- Ajax enabled JSF component library
 - 100+ components...
- Two tag libraries
 - a4j : Framework level ajax support
 - rich : Component level ajax support
- Skinning & Themes
- Component Development Kit (CDK)
- Dynamic Resource Framework

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- An Introduction
- Built in Ajax
- Built in Resource Handling
- Component Development
- Client Side Bean Validation
- Summary & Project Updates

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





JSF 2.0 Built in Ajax via Tags

```
<h:inputText id="score_input" value="#{gameBean.score}" />
    <f:ajax event="keyup" execute="@this" render="@form" />
</h:inputText>

<h:outputText id="score_output" value="#{gameBean.score}" />
```

Execute [**@this**]

id's to process on server

Render [**@none**]

id's to re-render to client

Key words

@all

@none

@this

@form

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





RichFaces makes it better with..

- a4j:region
- a4j:queue
- Partial updates for complex components

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



RichFaces Ajax Region



```
<!-- Verify Address as part of a large form -->
```

```
<a4j:region>
```

```
    <h:inputText id="addr_01" ... />
```

```
    <h:inputText id="addr_02" ... />
```

```
    <h:inputText id="city" ... />
```

```
    <h:inputText id="state" ... />
```

```
    <h:inputText id="zip" ... />
```

```
    ...
```

```
    <h:commandLink action="#{registerBean.verifyAddress}" >
```

```
        <a4j:ajax/>
```

```
        <!--
```

```
        <f:ajax execute="@this addr_01 addr_02 city state  
zip"/>
```

```
        -->
```

```
    </h:commandLink>
```

```
</a4j:region>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





JSF 2.0 Request Queue

- Specification added Ajax requests
- Requests need to be queued to preserve state
- Good, but limited
 - Very few options
 - No client side api
- Can lead to traffic jams...

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Request Queues

- Adds a logical queue on top
 - Usability & customizations
- Options
 - requestDelay & request grouping
 - IgnoreDupResponses (in development)
- Client side API
 - `<queueObj>.getSize()`
 - `<queueObj>.isEmpty()`
 - `<queueObj>.set/getQueueOptions()`
 - etc...

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Request Queues

```
<h:form id="form">
  <!-- Queue applied to whole form -->
  <a4j:queue requestDelay="500"/>

  <h:inputText id="team1" value="#{gameBean.team1}" />
    <a4j:ajax event="keyup" .../>
  </h:inputText>
  <h:inputText id="team2" value="#{gameBean.team2}" />
    <a4j:ajax event="keyup" .../>
  </h:inputText>
  ...
</h:form>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





RichFaces Request Queues

```
<!-- Player lookups are more costly, send less often -->
<a4j:queue name="player_queue" requestDelay="1000"/>
...
<h:inputText id="player" value="#{gameBean.cur_player}" />
  <a4j:ajax event="keyup" ...>
    <!-- Requests from here will use player_queue -->
    <a4j:attachQueue name="player_queue"/>
  </a4j:ajax>
</h:inputText>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





Partial Updates for Component Parts

- JSF 2.0 limits updates
 - Key words (@form, @this, etc..)
 - specific component ids
- RichFaces adds updates within complex components
 - Table supports: @body, @header, @footer
 - @row & @cell still in development
 - Support for collection of rows
 - Refer to specific rows for render
 - In development
 - Others, like tabPanel:@header in future

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Partial Updates for Iteration Components



```
<!-- Will render only the body of the table -->  
<a4j:ajax render="table_id:@body"/>
```

```
<!-- Will render only the current row -->  
<rich:column>  
    <a4j:commandButton render="@row"/>  
</rich:column>
```

```
<!-- Will render the collection of rows/cells -->  
<a4j:commandButton render=  
    "#{rich:rowKeys(someCollection, 'tableId')}:cellId"/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- An Introduction
- Built in Ajax
- Built in Resource Handling
- Component Development
- Client Side Bean Validation
- Summary & Project Updates

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





JSF 2.0 Loading Images

```
<!-- Loads from <web-root>/resources -->  
<h:graphicImage name="fenway.jpeg" />
```

```
<!-- Loads from stadiums.jar/META-INF/resources -->  
<h:graphicImage name="foxborough.jpeg" library="stadiums"/>
```

```
<!-- EL notation -->  
<h:graphicImage value="#{resource['stadiums:foxborough.jpeg']}" />
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





JSF 2.0 Loading JavaScript & CSS

- Two new resource components

```
<!-- Loads where you put it -->  
<h:outputScript name="patriots.js"/>
```

```
<!-- Loads in the <h:head> -->  
<h:outputScript name="patriots.js" target="head"/>
```

```
<!-- CSS always loads in <h:head> -->  
<h:outputStylesheet name="celtics.css" library="garden"/>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





What if you need to manipulate resources...

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Dynamic Resources in RichFaces

- Extends JSF 2.0 resource handler
- Used by RichFaces internally
 - Skinning
 - CSS with embedded EL
 - Java2D
- Make your own!!
 - Custom icons with state info
 - Dynamic CSS & JS



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





@DynamicResource

```
public class BradyResource implements UserResource{
    protected byte[] mossObject;
    protected byte[] welkerObject;

    protected boolean longBall;

    public InputStream getInputStream() throws IOException
    {
        byte[] data;
        if (longBall) {
            data = mossObject;
        } else{
            data = welkerObject;
        }
        return new ByteArrayInputStream(data);
    }
    ....
}
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Now Go Get It...



```
<h:outputScript library="patriots" name="BradyResource"/>
```

- Assuming BradyResource is in
 - Base package
 - patriots.jar

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Sending in the parameters...

```
@ResourceParameter  
protected boolean longBall;
```

```
<h:outputScript value="#{resource['patriots:  
BradyResource?longBall=true]}"/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- An Introduction
- Built in Ajax
- Built in Resource Handling
- **Component Development**
- Client Side Bean Validation
- Summary & Project Updates

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





JSF 2.0 Component Development

- New annotations
 - `@FacesComponent`, `@FacesRenderer`
- Create:
 - Tag handlers and tag lib files
 - `Faces-config.xml` file
 - Extend `UIComponentBase` class
 - Implement the `decode/encode*()` methods
 - Component functionality classe
 - Maintain it all...

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JSF 2 Component Development



```
ResponseWriter writer = context.getResponseWriter();
assert(writer != null);
String formClientId = RenderKitUtils.getFormClientId(command, context);
if (formClientId == null) {
    return;
}

//make link act as if it's a button using javascript
writer.startElement("a", command);
writeIdAttributeIfNecessary(context, writer, command);
writer.writeAttribute("href", "#", "href");
RenderKitUtils.renderPassThruAttributes(context, writer, command, ATTRIBUTES,
    getNonOnClickBehaviors(command));

RenderKitUtils.renderXHTMLStyleBooleanAttributes(writer, command);

String target = (String) command.getAttributes().get("target");
if (target != null) {
    target = target.trim();
} else {
    target = "";
}

Collection<ClientBehaviorContext.Parameter> params = getBehaviorParameters(command);
RenderKitUtils.renderOnClick(context, command, params, target, true);

writeCommonLinkAttributes(writer, command);

// render the current value as link text.
writeValue(command, writer);
writer.flush();
```

**Just part of
CommandLinkRenderer.java**

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



RichFaces Component Development Kit



- Does much of the leg work for you!
- In many situations you only need:
 - Abstract component class
 - Renderer template in xml/xhtml
- Integrates into your build
 - Maven-plugin
- IDE support
 - standard xml files & schema

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Renderer Template

```
<cc:implementation>
  <c:choose>
    <c:when test="#{not component.attributes['disabled']}">
      <a id="#{clientId}" name="#{clientId}"
        cdk:passThroughWithExclusions="value"
        onclick="#{this.getOnClick(facesContext, component)}"
        href="#">
        #{component.attributes['value']}
        <cdk:call expression="renderChildren(facesContext, component)" />
      </a>
    </c:when>
    <c:otherwise>
      <span id="#{clientId}" cdk:passThroughWithExclusions="value">
        #{component.attributes['value']}
        <cdk:call expression="renderChildren(facesContext, component)" />
      </span>
    </c:otherwise>
  </c:choose>
</cc:implementation>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Component Development

- Abstract component class
 - Only implement component functionality
 - CDK will extend it based on:
 - Template
 - Config options
- CDK will compile template for renderer
 - Including concrete component class
 - Getters/setters, etc..
 - Tag handlers, tab lib, and faces-config files

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- An Introduction
- Built in Ajax
- Built in Resource Handling
- Component Development
- **Client Side Bean Validation**
- Summary & Project Updates

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





What is Bean Validation

- Bean Validation JSR-303
 - Part of Java EE6
- Generic, tier independent constraints
- Custom, and composition constraints possible
- Constraint groups, and object graph validation too
- Message localization
- Define constraints once, apply everywhere*

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

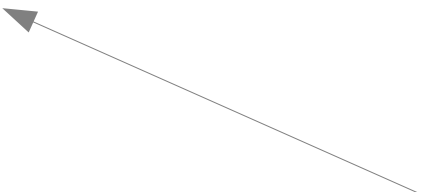




Bean Validation Sample

```
public class User {  
    @NotEmpty @Size(max=100)  
    private String loginname;  
  
    @NotEmpty  
    @Pattern(regexp = "[a-zA-Z0-9]+@[a-zA-Z0-9]+\\. [a-zA-Z0-9]+")  
    private String email;  
  
    @NotEmpty @ValidTeam  
    private String team;  
}
```

Custom constraint



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JSF 2.0 Bean Validation Integration



Login:

```
<h:inputText id="login" value="#{user.login}"/>
<h:message for="login"/>
<br/>
```

Email:

```
<h:inputText id="email" value="#{user.email}"/>
<h:message for="email"/>
<br/>
```

Team:

```
<h:inputText id="team" value="#{user.team}">
    <f:validateBean disabled="true"/>
</h:inputText>
<h:message for="team"/>
<br/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





RichFaces Bean Validation

- **Takes it to the Client**
 - Ajax fallback options
- JavaScript:
 - Validation, Converters, Messages
- Can be applied to:
 - Inputs, Forms, View, or Application wide
- Develop you own validators
 - Client/server JS mapping

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Client Validation - Inputs

Login:

```
<h:inputText id="login" value="#{user.login}">
    <rich:clientValidator event="keyup"/>
</h:inputText>
<h:message for="login"/>
<br/>
```

Email:

```
<h:inputText id="email" value="#{user.email}">
    <rich:clientValidator event="keyup"/>
</h:inputText>
<h:message for="email"/>
<br/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



RichFaces Client Validation - Forms



```
<h:form id="register">
  <!-- This will verify when form submitted
       & default JSF event of form children -->
  <rich:clientValidator/>
```

Login:

```
<h:inputText id="login" value="#{user.login}"/>
<h:message for="login"/>
<br/>
```

Email:

```
<h:inputText id="email" value="#{user.email}"/>
<h:message for="email"/>
<br/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





RichFaces Client Validation

```
<h:form id="register">  
  <rich:clientValidator/>
```

Login:

```
<h:inputText id="login" value="#{user.login}">  
  <!-- would trigger validation pre-request on key up-->  
  <rich:clientValidator event="keyup"/>  
  <f:ajax event="keyup" listener="#{bean.loginTaken}"/>  
</h:inputText>  
<h:message for="login"/>
```

...

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





RichFaces Client Validation

```
<h:form id="register">
  <rich:clientValidator/>
  ...
  Name:
  <h:inputText id="name" value="#{user.name}">
    <!-- Disable client validation -->
    <rich:clientValidator disabled="true"/>
  </h:inputText>
  <h:message for="name"/>
  ...
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





Beyond JSF 2.0 with RichFaces

- An Introduction
- Built in Ajax
- Built in Resource Handling
- Component Development
- Client Side Bean Validation
- [Summary & Project Updates](#)

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JSF 2.0 + RichFaces == Rich & Easy JSF



- JSF 2.0 added a lot
 - Left the door open for inovation!!
- RichFaces is prototyping the future of the spec
 - You get to use it first with RichFaces
- Improvements
 - Usability, and performance
 - Dynamic resources
 - Component development
 - Client side bean validation
 - & more....

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Rich Components!



Edit **Find** **Filter**

Flag	State Name ↓	State Capital ↓	Time Zone
	Maine	Augusta	GMT-5
	Maryland	Annapolis	GMT-5
	Massachusetts	Boston	GMT-5

Available Items

Open
 Save
 Save All
 Delete

Copy all

Copy

Remove

Remove All

Currently Active Items

Create Folder
 Create Doc
 Find

First

Up

Down

Last

<< < June, 2010 > >> x

Sun	Mon	Tue	Wed	Thu	Fri	Sat
23 30	31	1	2	3	4	5
24 6	7	8	9	10	11	12
25 13	14	15	16	17	18	19
26 20	21	22	23	24	25	26
27 27	28	29	30	1	2	3
28 4	5	6	7	8	9	10

Today Apply

File Links

New

Open

Save As...

Close

Exit

Save

Save All

a

	Atlanta	Georgia
	Augusta	Maine
	Annapolis	Maryland
	Albany	New York
	Austin	Texas

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





RichFaces Project Updates

- 4.0 Release Plans
 - Alpha2 just released
 - Milestone releases every month
- Lots of ways to get involved and stay informed
 - Twitter (<http://twitter.com/richfaces>)
 - Forums
 - <http://community.jboss.org/en/richfaces/dev>
 - Development projects & tasks
 - <https://jira.jboss.org/browse/RF>
 - Team Meetings, and IRC
 - #richfaces @ irc.freenode.net

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- Conversion & Validation
- Page Actions
- Messages
- Empowered Testability
- Summary

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Seam Faces – An Introduction

- Portable extension for JavaServer Faces & CDI
- One of many independent modules in Seam 3
- Plug & Play
- The icing on your cake

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- **Conversion & Validation**
- Page Actions
- Messages
- Empowered Testability
- Summary

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Conversion/Validation in JSF 2.0

- No more XML

```
@FacesValidator("addressValidator")
public class AddressValidator implements Validator
{
    ...
}
```

- Automatic conversion

```
@FacesConverter(forClass = ZipCode.class)
public class ZipCodeConverter implements Converter
{
    ...
}
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Cross-field Validation in JSF 2.0

```
@FacesValidator("badAddressValidator")
public class BadAddressValidator implements Validator
{
    Directory directory
        = new DatabaseAddressDirectory("host", 3036, "schema", "username", "password");

    @Override
    public void validate(FacesContext context, UIComponent c, Object val) throws ValidatorException
    {
        String city = context.getExternalContext().getRequestParameterMap().get("form:table:city");
        String state = context.getExternalContext().getRequestParameterMap().get("form:table:state");
        String zipValue = context.getExternalContext().getRequestParameterMap().get("form:table:zip");

        ZipCode zip = new ZipCode(Integer.valueOf(zipValue));

        if (directory.exists(city, state, zip))
        {
            throw new ValidatorException(new FacesMessage("Invalid address. Please try again.));
        }
    }
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





The Seam Faces Answer

```
@FacesValidator("addressValidator")
public class AddressValidator implements Validator
{
    @Inject Directory directory;

    @Inject
    @InputField String city;

    @Inject
    @InputField String state;

    @Inject
    @InputField ZipCode zip;

    public void validate(FacesContext context, UIComponent c, Object val) throws ValidatorException
    {
        if (!directory.exists(city, state, zip))
        {
            throw new ValidatorException("Invalid address. Please try again.");
        }
    }
}
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





The Seam Faces Answer

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:s="http://jboss.org/seam/faces">

  <h:form id="form">

    City: <h:inputText id="city" value="#{bean.city}" /> <br/>
    State: <h:inputText id="state" value="#{bean.state}" /> <br/>
    Zip: <h:inputText id="zipCode" value="#{bean.zip}" /> <br/>

    <h:commandButton id="submit" value="Submit" action="#{bean.submit}" />

    <s:validateForm validatorId="addressValidator" fields="zip=zipCode" />

  </h:form>

</html>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Seam Faces - Converters & Validators

- `@Inject` and `@*Scoped` support
- Automatic type-conversion
- Built-in support for `i18n` and `l10n`
- Designed for re-use

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- Conversion & Validation
- **Page Actions**
- Messages
- Empowered Testability
- Summary

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Page actions with GET in JSF 2.0

```
<f:metadata>
  <f:viewParam name="id" value="#{blogManager.entryId}" />
  <f:event name="preRenderView" listener="#{blogManager.loadEntry}" />
</f:metadata>
```

```
@ManagedBean
@RequestScoped
public class BlogManager
{
    public void listener(ComponentSystemEvent event)
        throws AbortProcessingException
    {
        // do something here
    }
}
```

“You can do that?”

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Page actions with GET in JSF 2.0

- But can you navigate?
- Can you execute in a different phase, conditionally?
- Can you enable/disable execution on a POST?

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Seam Faces Answer



```
<f:metadata>
  <f:viewParam name="id" value="#{blogManager.entryId}" />

  <s:viewAction action="#{blogManager.loadEntry}" phase="APPLY_REQUEST_VALUES"
    if="#{conversation.transient}" onPostback="true" />
</f:metadata>
```

```
@Named
public class BlogManager
{
    private String entryId;
    private Entry entry;

    public void loadEntry()
    {
        // attempt to load the entry
    }
}
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Seam Faces Answer



```
<navigation-rule>
  <from-view-id>/entry.xhtml</from-view-id>
  <navigation-case>
    <from-action>#{blogManager.loadEntry}</from-action>
    <if>#{empty blogManager.entry}</if>
    <to-view-id>/home.xhtml</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Seam Faces – Page actions with GET

- Integrated with Faces Navigation
- Invoke at specific phases of JSF lifecycle
- Conditional invocation on GET / POST
- Conditional invocation via EL

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- Conversion & Validation
- Page Actions
- [Messages](#)
- Empowered Testability
- Summary

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Faces Messages in JSF 2.0



```
FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(SEVERITY_INFO, "Finally...", null));
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Faces Messages after a Redirect



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Seam Faces Answer



```
@Inject  
Messages messages;
```

```
messages.info("What a breath of {0} {1}!", "fresh", "air");
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





The Seam Faces Answer

- A true @FlashScoped context
- Lives from “now” until Render Response (even through HTTP redirects and navigation)
- Messages automatically saved in flash scope
- Fluent, testable API from [Seam International](#) Module
- Supports i18n and resource_bundle.properties

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- Conversion & Validation
- Page Actions
- Messages
- [Empowered Testability](#)
- Summary

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Testing JSF 2.0



```
FacesContext.getCurrentInstance();
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



The Seam Faces Answer



```
@Inject  
FacesContext context;
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Testing with Seam Faces

- Loose-coupling through @Inject
- Providers for:

@Inject FacesContext

@Inject ExternalContext

@Inject NavigationHandler

@Inject @Faces Locale

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Beyond JSF 2.0 with Seam Faces

- An Introduction
- Conversion & Validation
- Page Actions
- Messages
- Empowered Testability
- [Summary](#)

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Summary of JSF 2.0 with Seam Faces

- Conversion & Validation
 - `@Inject` and `@*Scoped` support
 - Cross-field validation
- Page Actions
 - With integrated navigation
- Messages
 - Translated, Survive redirects via `@FlashScoped`
- Empowered Testability
 - Avoid statics, use `@Inject`

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Seam Faces Plan

- Release Plans:
 - Working on 3.0.0.Alpha4
 - Beta by late-July
 - [Try it](#) now!
- Get involved in the Project:
<http://seamframework.org/Seam3/FacesModule>
- Submit feature requests (and bugs) to Seam Faces:
<https://jira.jboss.org/browse/SEAMFACES>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Whats bugging you with JSF 2.0

- What do you wish were added to spec?
- What is still missing?
- We are maintaining a list of updates for the next rev
<http://seamframework.org/Documentation/JSF21>
 - Both Seam, and RichFaces topics

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT





Wrap up and Q&A

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

