JSF IN THE MODERN AGE

KCDC Presentation
Friday, May 16, 2014 at 3:20 p.m.

Keith Shakib
Principal Consultant
Keyhole Software

Titanium Sponsors

AJi

VALOREM CONSULTING
WWW.VALOREMCONSULTING.COM
REGISTER TO WIN AN XBOX!

PAIGE TECHNOLOGIES
INTELLIGENT PAIRING. PERPETUAL SUCCESS.

AdventureTech
www.adventuretechgroup.com

Platinum Sponsors

ADAPTIVE SOLUTIONS GROUP

DST SYSTEMS

Balance Innovations

Sprint

epiQ SYSTEMS

DevExpress

jack henry & ASSOCIATES INC.

O'REILLY

Kauffman LABS for Enterprise Creation

FREIGHTQUOTE

Gold Sponsors

Cerner

CENTRIQ TRAINING

Bradford & Galt CONSULTING SERVICES

ADVANTAGE TECH

KEYHOLE SOFTWARE

ComponentOne a division of GrapeCity

dsi

LRS Consulting Services

GARMIN

JetBRAINS

Microsoft

MULTI SERVICE INNOVATION WHERE IT MATTERS.

NETCHEMIA Transforming the way education works

OAKWOOD

stackify

KU EDWARDS CAMPUS The University of Kansas

perceptivesoftware from Lexmark

TEKsystems

twilio CLOUD COMMUNICATIONS

UNITEDLEX

JSF – WHAT IS IT?

"JavaServer Faces (JSF) is a Java specification for building component-based user interfaces for web applications. It was formalized as a standard through the Java Community Process and is part of the Java Platform, Enterprise Edition."

– according to Wikipedia

JAVASERVER FACES 1.0

JSF 1.0 – spec released 3/11/2004

From JavaWorld Nov 29, 2002
by David Geary

"In a nutshell, JSF eases Web-based application development because it:

- Allows you to create user interfaces from a set of standard, reusable server-side components
- Provides a set of JSP tags to access those components
- Transparently saves state information and repopulates forms when they re-display
- Provides a framework for implementing custom components
- Encapsulates event handling and component rendering so you can use standard JSF components or custom components to support markup languages other than HTML
- Allows tool vendors to develop IDEs for a standard web application framework"

JAVASERVER FACES 1.2

JSF 1.2 – 5/11/2006 (JEE5)

From roseindia.net May 11, 2006

1. Unified Expression Language(EL)
2. AJAX Support
3. New Tree Creation and Content Interweaving Model for Faces applications that use JSP
4. Integration with JSTL
5. Back Button issues and Multi-Frame or Multi-Window Faces Apps
6. Associating a message with a particular component in the page
7. Expose an application-wide ResourceBundle to the EL
8. Use of multiple renderKits
9. Provide XML Schema for the config files, instead of using DTD
10. Security enhancements for client side state saving
11. Solve the "duplicate button press" problem
12. Portlet-related bug fixes

JAVASERVER FACES 2.0

JSF 2.0 – July 1, 2009 (JEE6)

From codedairy.com July 18, 2012
by Kuldip Bajwa

1. AJAX support within tags
2. Composite components with listeners, attributes, and events
3. Partial state saving
4. View parameters
5. Navigation enhancements including bookmark-ability, navigation with XML navigation rules, and conditional navigation
6. Exception handling
7. Expression language enhancements
8. Validation with JSR-303 Bean Validation
9. New scopes including custom
10. FacesContext servlet access
11. Annotations may obviate the need for XML

JAVASERVER FACES 2.2

JSF 2.2 – May 21, 2013

from http://jdevelopment.nl/jsf-22/#220
Sept 1, 2012 by Arjan Tijms

… the proposed final list of Big Ticket Features became:

- Sensible HTML5 support
- Resource Library Contracts (a toned-down Multi-Templating)
- Faces flows
- Stateless views

JSF EXAMPLE

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Welcome to JBoss!

You have successfully deployed a Java EE 6 Enterprise Application.

Learn more about JBoss Enterprise Application Platform 6.

- Documentation
- Product Information

Member Registration

Enforces annotation-based constraints defined on the model class.

Name:

Email:

Phone #:

Register

Members

Id	Name	Email	Phone #	REST URL
0	John Smith	john.smith@mailinator.com	2125551212	/rest/members/0

REST URL for all members: /rest/members

This project was generated from a Maven archetype from JBoss.

JSF EXAMPLE

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Welcome to JBoss!

You have successfully deployed a Java EE 6 Enterprise Application.

Learn more about JBoss Enterprise Application Platform 6.

- Documentation
- Product Information

Member Registration

Enforces annotation-based constraints defined on the model class.

Name: Keith Shakib

Email: kshakib@keyholesoftware.com

Phone #: 8168375309

Register

Members

Id	Name	Email	Phone #	REST URL
0	John Smith	john.smith@mailinator.com	2125551212	/rest/members/0

REST URL for all members: /rest/members

This project was generated from a Maven archetype from JBoss.

JSF EXAMPLE

RED HAT' JBOSS'
ENTERPRISE
APPLICATION PLATFORM

Welcome to JBoss!

You have successfully deployed a Java EE 6 Enterprise Application.

Learn more about JBoss Enterprise Application Platform 6.

- Documentation
- Product Information

Member Registration

Enforces annotation-based constraints defined on the model class.

Name:

Email:

Phone #:

Register

- Registered!

Members

Id	Name	Email	Phone #	REST URL
0	John Smith	john.smith@mailinator.com	2125551212	/rest/members/0
1	Keith Shakib	kshakib@keyholesoftware.com	8168375309	/rest/members/1

REST URL for all members: /rest/members

This project was generated from a Maven archetype from JBoss.

JSF EXAMPLE

RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Welcome to JBoss!

You have successfully deployed a Java EE 6 Enterprise Application.

Learn more about JBoss Enterprise Application Platform 6.

- Documentation
- Product Information

Member Registration

Enforces annotation-based constraints defined on the model class.

Name: size must be between 1 and 25

Email:

Phone #: numeric value out of bounds (<12 digits>.<0 digits> expected)

Register

Members

Id	Name	Email	Phone #	REST URL
0	John Smith	john.smith@mailinator.com	2125551212	/rest/members/0
1	Keith Shakib	kshakib@keyholesoftware.com	8168375309	/rest/members/1

REST URL for all members: /rest/members

This project was generated from a Maven archetype from JBoss.

JSF EXAMPLE CODE
default.xhtml

default.xhtml

```
 2     JBoss, Home of Professional Open Source.
17  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
18      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
19  <html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html" xmlns:ui="http://java.sun.com/jsf/facelets">
20  <h:head>
21      <title>${artifactId}</title>
22      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
23      <h:outputStylesheet name="css/screen.css" />
24  </h:head>
25  <h:body>
26      <div id="container">
27          <div class="dualbrand">
28              <img src="resources/gfx/rhjb_eap_logo.png" />
29          </div>
30          <div id="content">
31              <ui:insert name="content">
32                      [Template content will be inserted here]
33              </ui:insert>
34          </div>
35          <div id="aside">
36              <p>Learn more about JBoss Enterprise Application Platform 6.</p>
37              <ul>
38                  <li><a href="https://access.redhat.com/site/documentation/JBoss_Enterprise_Application_Platform/">Documentation</a></li>
39                  <li><a href="http://red.ht/jbeap-6">Product Information</a></li>
40              </ul>
41          </div>
42          <div id="footer">
43              <p>
44                  This project was generated from a Maven archetype from JBoss.<br />
45              </p>
46          </div>
47      </div>
48  </h:body>
49  </html>
50
```

JSF EXAMPLE CODE
index.xhtml (PART I)

index.xhtml

```
1  <?xml version="1.0" encoding="UTF-8"?>
3      JBoss, Home of Professional Open Source
18 <ui:composition xmlns="http://www.w3.org/1999/xhtml"
19     xmlns:ui="http://java.sun.com/jsf/facelets"
20     xmlns:f="http://java.sun.com/jsf/core"
21     xmlns:h="http://java.sun.com/jsf/html"
22     template="/WEB-INF/templates/default.xhtml">
23     <ui:define name="content">
24         <h1>Welcome to JBoss!</h1>
25
26         <div>
27             <p>You have successfully deployed a Java EE 6 Enterprise
28                 Application.</p>
29         </div>
30
31         <h:form id="reg">
32             <h2>Member Registration</h2>
33             <p>Enforces annotation-based constraints defined on the
34                 model class.</p>
35             <h:panelGrid columns="3" columnClasses="titleCell">
36                 <h:outputLabel for="name" value="Name:" />
37                 <h:inputText id="name" value="#{newMember.name}" />
38                 <h:message for="name" errorClass="invalid" />
39
40                 <h:outputLabel for="email" value="Email:" />
41                 <h:inputText id="email" value="#{newMember.email}" />
42                 <h:message for="email" errorClass="invalid" />
43
44                 <h:outputLabel for="phoneNumber" value="Phone #:" />
45                 <h:inputText id="phoneNumber"
46                     value="#{newMember.phoneNumber}" />
47                 <h:message for="phoneNumber" errorClass="invalid" />
48             </h:panelGrid>
49
50             <p>
51                 <h:panelGrid columns="2">
52                     <h:commandButton id="register"
53                         action="#{memberController.register}"
54                         value="Register" styleClass="register" />
55                     <h:messages styleClass="messages"
56                         errorClass="invalid" infoClass="valid"
57                         warnClass="warning" globalOnly="true" />
58                 </h:panelGrid>
59             </p>
60         </h:form>
```

JSF EXAMPLE CODE
index.xhtml (PART 2)

index.xhtml

```
60          </h:form>
61          <h2>Members</h2>
62          <h:panelGroup rendered="#{empty members}">
63              <em>No registered members.</em>
64          </h:panelGroup>
65          <h:dataTable var="_member" value="#{members}"
66              rendered="#{not empty members}"
67              styleClass="simpletablestyle">
68              <h:column>
69                  <f:facet name="header">Id</f:facet>
70                  #{_member.id}
71              </h:column>
72              <h:column>
73                  <f:facet name="header">Name</f:facet>
74                  #{_member.name}
75              </h:column>
76              <h:column>
77                  <f:facet name="header">Email</f:facet>
78                  #{_member.email}
79              </h:column>
80              <h:column>
81                  <f:facet name="header">Phone #</f:facet>
82                  #{_member.phoneNumber}
83              </h:column>
84              <h:column>
85                  <f:facet name="header">REST URL</f:facet>
86                  <a
87                      href="#{request.contextPath}/rest/members/#{_member.id}">/rest/members/#{_member.id}</a>
88              </h:column>
89              <f:facet name="footer">
90              REST URL for all members: <a
91                      href="#{request.contextPath}/rest/members">/rest/members</a>
92              </f:facet>
93          </h:dataTable>
94      </ui:define>
95  </ui:composition>
```

JSF EXAMPLE CODE
Member.java

Member.java

```
 2  * JBoss, Home of Professional Open Source
17 package com.khs.jmaw.model;
18
19 import java.io.Serializable;
20
21 import javax.persistence.Column;
22 import javax.persistence.Entity;
23 import javax.persistence.GeneratedValue;
24 import javax.persistence.Id;
25 import javax.persistence.Table;
26 import javax.validation.constraints.Digits;
27 import javax.validation.constraints.NotNull;
28 import javax.validation.constraints.Pattern;
29 import javax.validation.constraints.Size;
30
31 @Entity
32 @Table
33 public class Member implements Serializable {
34
35     private static final long serialVersionUID = 1L;
36
37     @Id
38     @GeneratedValue
39     private Long id;
40
41     @NotNull
42     @Size(min = 1, max = 25)
43     @Pattern(regexp = "[^0-9]*", message = "Must not contain numbers")
44     private String name;
45
46     @NotNull
47     private String email;
48
49     @NotNull
50     @Size(min = 10, max = 12)
51     @Digits(fraction = 0, integer = 12)
52     @Column(name = "phone_number")
53     private String phoneNumber;
54
55     public String getEmail() {
56         return email;
57     }
```

JSF EXAMPLE CODE
MemberController.java

MemberController.java

```java
 * JBoss, Home of Professional Open Source
package com.khs.jmaw.controller;

import javax.annotation.PostConstruct;

// The @Model stereotype is a convenience mechanism to make this a request-scoped bean that has an
// EL name
// Read more about the @Model stereotype in this FAQ:
// http://sfwk.org/Documentation/WhatIsThePurposeOfTheModelAnnotation
@Model
public class MemberController {

    @Inject
    private FacesContext facesContext;

    @Inject
    private MemberRegistration memberRegistration;

    @Produces
    @Named
    private Member newMember;

    @PostConstruct
    public void initNewMember() {
        newMember = new Member();
    }

    public void register() throws Exception {
        try {
            memberRegistration.register(newMember);
            FacesMessage m = new FacesMessage(FacesMessage.SEVERITY_INFO, "Registered!", "Registration successful");
            facesContext.addMessage(null, m);
            initNewMember();
        } catch (Exception e) {
            String errorMessage = getRootErrorMessage(e);
            FacesMessage m = new FacesMessage(FacesMessage.SEVERITY_ERROR, errorMessage, "Registration unsuccessful");
            facesContext.addMessage(null, m);
        }
    }
```

JSF EXAMPLE CODE MemberRegistration.java (SERVICE)

MemberRegistration.java

```
 2⊕ * JBoss, Home of Professional Open Source..
17 package com.khs.jmaw.service;
18
19⊕ import com.khs.jmaw.model.Member;..
26
27 // The @Stateless annotation eliminates the need for manual transaction demarcation
28 @Stateless
29 public class MemberRegistration {
30
31⊖     @Inject
32      private Logger log;
33
34⊖     @Inject
35      private EntityManager em;
36
37⊖     @Inject
38      private Event<Member> memberEventSrc;
39
40⊖     public void register(Member member) throws Exception {
41          log.info("Registering " + member.getName());
42          em.persist(member);
43          memberEventSrc.fire(member);
44      }
45 }
46
```

JSF EXAMPLE CODE
MemberListProducer.java

MemberListProducer.java

```
 2  * JBoss, Home of Professional Open Source
17 package com.khs.jmaw.data;
18
19 import javax.annotation.PostConstruct;
29
30 @RequestScoped
31 public class MemberListProducer {
32
33     @Inject
34     private MemberRepository memberRepository;
35
36     private List<Member> members;
37
38     // @Named provides access the return value via the EL variable name "members" in the UI (e.g.
39     // Facelets or JSP view)
40     @Produces
41     @Named
42     public List<Member> getMembers() {
43         return members;
44     }
45
46     public void onMemberListChanged(@Observes(notifyObserver = Reception.IF_EXISTS) final Member member) {
47         retrieveAllMembersOrderedByName();
48     }
49
50     @PostConstruct
51     public void retrieveAllMembersOrderedByName() {
52         members = memberRepository.findAllOrderedByName();
53     }
54 }
55
```

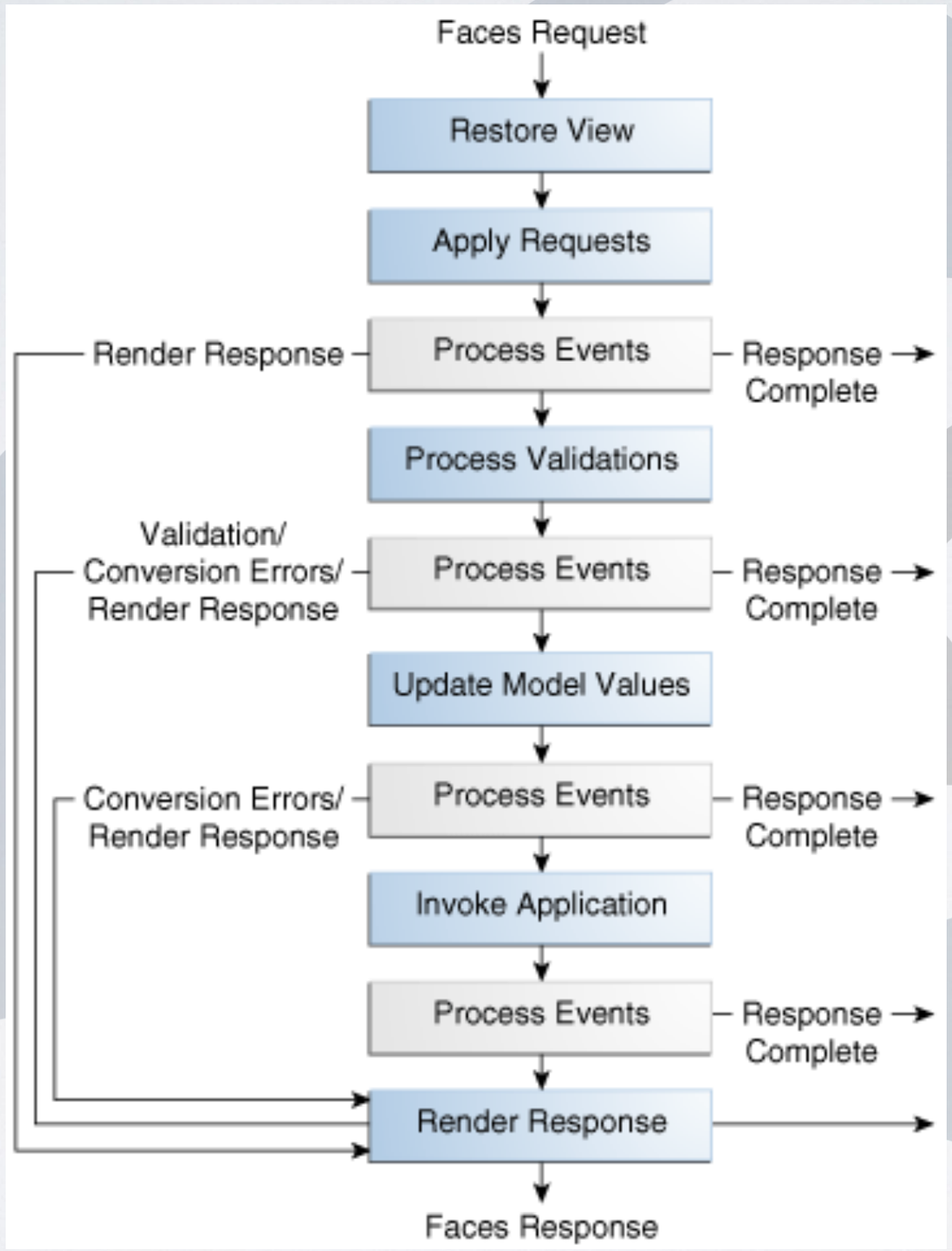
JSF EXAMPLE CODE
MemberResourceRESTService.java

MemberRepository.java | MemberResourceRESTService.java

```
 2  * JBoss, Home of Professional Open Source
17  package com.khs.jmaw.rest;
18
19  import java.util.HashMap;
46
48  * JAX-RS Example
52  @Path("/members")
53  @RequestScoped
54  public class MemberResourceRESTService {
55
56      @Inject
57      private Logger log;
58
59      @Inject
60      private Validator validator;
61
62      @Inject
63      private MemberRepository repository;
64
65      @Inject
66      MemberRegistration registration;
67
68      @GET
69      @Produces(MediaType.APPLICATION_JSON)
70      public List<Member> listAllMembers() {
71          return repository.findAllOrderedByName();
72      }
73
74      @GET
75      @Path("/{id:[0-9][0-9]*}")
76      @Produces(MediaType.APPLICATION_JSON)
77      public Member lookupMemberById(@PathParam("id") long id) {
78          Member member = repository.findById(id);
79          if (member == null) {
80              throw new WebApplicationException(Response.Status.NOT_FOUND);
81          }
82          return member;
83      }
84
```

```
[{"id":0,"name":"John
Smith","email":"john.smith@mailinator.com","phoneNumber":"2125551212"},
{"id":1,"name":"Keith
Shakib","email":"kshakib@keyholesoftware.com","phoneNumber":"8168375309"}]
```

JSF LIFE CYCLE



JSF {NOT SO} ADVANCED TOPICS

- **Navigation**
  - faces-config.xml
  - Implicit navigation
  - outcome= attribute
- **Component Libraries**
  - PrimeFaces
  - RichFaces
  - …Faces
- **AJAX / JavaScript**
- **Adaptive Controls**

TECHNOLOGY RADAR

Radar home Techniques Tools Platforms Languages & frameworks Radar A-Z

JSF

•Hold
January 2014

We continue to see teams run into trouble using JSF -- JavaServer Faces -- and are recommending you avoid this technology. Teams seem to choose JSF because it is a J2EE standard without really evaluating whether the programming model suits them. We think JSF is flawed because it tries to abstract away HTML, CSS and HTTP, exactly the reverse of what modern web frameworks do. JSF, like ASP.NET webforms, attempts to create statefulness on top of the stateless protocol HTTP and ends up causing a whole host of problems involving shared server-side state. We are aware of the improvements in JSF 2.0, but think the model is fundamentally broken. We recommend teams use simple frameworks and embrace and understand web technologies including HTTP, HTML and CSS.

About ThoughtWorks

We are a software company and a community of passionate, purpose-led individuals. We think disruptively to deliver technology to address our clients' toughest challenges, all while seeking to revolutionize the IT industry and create positive social change.

Connect with us

Sign up for our perspectives newsletter

http://www.thoughtworks.com/radar/#/languages-and-frameworks/683

JSF - WHAT ARE THE ALTERNATIVES?

Java —> JavaScript, HTML, CSS
Server —> Client-side rendering
Faces —> JavaScript Component Frameworks
{Angular/Backbone/Ember/etc}

But why ?

Responsive/Adaptive Design
Offload Rendering to Client
Highly de-coupled From Server
User Interface Focus

So why not?

Target Users
Technology Maturity
Developer Skillset

SO, WHY SHOULDN'T I BE MOVING AWAY FROM JSF, TOO?

Target Users – It takes extra time and design consideration to plan ahead for supporting multiple types of devices; is this a worthwhile trade-off?

Technology Maturity - Which JavaScript framework do you choose? Where is the industry going?

Developer Skillset - What will the ramp-up time be for the new frameworks? It is a paradigm shift. What if you have Java developers who are not efficient in JavaScript?
Keep in mind your JavaScript framework developers need to be well-disciplined to prevent future maintenance nightmares.

SO, IS JSF RIGHT FOR MY ORGANIZATION?

Recommendations:

- Assess where your applications will be running in the next 2-10 years
- Think about how your organization deals with technology risk
- Ask yourself if now is the right time to develop new User Interface skills in your organization
- Do not be afraid to dip your toes in the water. For instance, deploying to both JSF pages and AngularJS pages is okay!

DOWNLOAD THIS PRESENTATION

Presentation materials available for download on the Keyhole GitHub:

bit.ly/keyholekcdc14

Other Keyhole Presentations Available:

1. Grunt 101
2. Java RESTful End Point Frameworks to Support the Shift to HTML5 SPA
3. Advanced JavaScript Debugging Techniques for Agile Teams

Stop by Keyhole Software's booth to talk about career opportunities with my team & get entered to win an HP Chromebook 14.