



The Power of JSF 2.0

- Sieben Thesen und Antithesen -

SENS

Andy Bosch

Freier Berater und Trainer

Betreiber von www.jsf-forum.de

SENS-Experte (www.SoftwareExperts.de)

Was habe ich in den nächsten 45 Minuten vor

- JSF 2.0 vorstellen und damit vor allem demonstrieren, wie umfangreich und mächtig der aktuelle Standard ist.
- Bestehende Vorurteile ausräumen ;-)
- Die Idee der Standardisierung und wie sie gelebt wird (bzw. gelebt werden kann) näherbringen

Entstehungsgeschichte

- JSF 1.0 wurde im Sommer 2004 veröffentlicht
- JSF 1.2: Kleineres Update im Mai 2006
Fokus: Alignment mit der JSP EL → Unified EL
- JSF 2.0 (JSR-314) begonnen im Sommer 2007
Spec Lead: Ed Burns und Roger Kitain
- Aktueller Status: Spec ist „fertig“ (seit Sommer 2009)
Mojarra und Apache MyFaces haben eine Implementierung

Wie entstand die Spezifikation?

- Offene Diskussion
- Einholen von Ideen aus dem OpenSource Bereich
- Keine Lösung entwickelt im Elfenbeinturm
- Sehr aktive, kommunikative Expert Group
- Viele bekannte Namen, die vormals Lösungen im JSF-Umfeld erstellt haben, z.B.
 - Gavin King (Seam)
 - Alexandr Smirnov (Rich Faces)
 - Ted Goddard (ICEFaces)
 - Ken Paulson (JSF Templating).
 - Jacob Hookom (Facelets)
 - ...

Tolle neue Feature-Welt ...

Ajax

Validation

Resources

Annotations

Config Annotations

Composite Components

Facelets

System Events

Project Stage

New Scopes

Behauptung 1: „JSF Setups sind zu komplex“

- Ne, das stimmt so nicht
 - Wir haben mittlerweile:
 - Managed Beans über Annotation
 - implizite Navigation
 - Basis ist Xhtml / Facelets
 - Künftig Servlet 3.0-Vereinfachungen

Wie setzte ich ein Projekt auf?

- Einfach ein Webprojekt anlegen
(Libraries sind Teil des Servers / Java EE Stacks)
- Eintrag in web.xml (entfällt mit Servlet 3.0)
- Loslegen



VDL – View Declaration Language

Standardization and improvement of Facelets.
We could call it Facelets 2.0.

- Kernfunktionalität von VDL wird geerbt von Facelets und JSFTemplating
- Facelets wird in den Standard als bevorzugte Viewtechnologie erhoben.
- JSP wird als deprecated gekennzeichnet.
- Bei der „Erfindung“ von Facelets sollte eine Viewtechnologie entstehen, die ausschließlich auf JSF fokussiert sein soll.

Managed Beans mit Annotations

Klassisch:

```
<managed-bean>
  <managed-bean-name>SimpleBean</managed-bean-name>
  <managed-bean-class>
    de.jsf.SimpleBean
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
```

Neu:

```
@ManagedBean
@SessionScoped
public class SimpleBean {
    ...
}
```

Vereinfachte Navigation

- Es gibt auch die Möglichkeit, per Konvention zu navigieren.

```
<h:commandButton value=„Zeige Detail“ action=„detail“/>
```

→ Suche nach **detail.xhtml** im gleichen Verzeichnis

Auch möglich:

```
<h:commandButton value=„Zeige Detail“  
  action=„/modull/detail“/>
```

```
<h:commandButton value=„Zeige Detail“  
  action=„/modull/detail?faces-redirect=true“ />
```

Behauptung widerlegt



Behauptung 2: „JSF verbraucht viel Hauptspeicher“

- Ne, das stimmt so auch nicht
 - Früher: Nicht-optimiertes State Saving
 - Heute: Partial State Saving
 - Tuning des State Savings dennoch notwendig
 - JSF hat aber einen State (und das ist auch gut so)
 - Für sex.com ist JSF nicht bestimmt

Behauptung widerlegt



Behauptung 3: „JSF 2 kann ja nicht einmal Html5“

- Sorry, aber auch wieder falsch
- JSF kann heute schon mit Html 5 zusammenarbeiten



Was ist eigentlich Html5?

- Html5 ist die nächste Version von Html, XHTML und Html DOM
- Initiiert durch das W3C und dem WHATWG (Web Hypertext Application Technology Working Group)
- Aktuell sind Drafts bekannt und teilweise auch schon in Browsern umgesetzt
- Html 5 = Html + JavaScript + Css
- Zieltermin (Gerüchte): 2012

Was kann Html5?

- Unterstützung von Video und Audio
- Offline Storage ("Super Cookies")
- Neue Formelemente wie Datum, Range, Zahl, E-Mail ...
- Canvas Element für Drawing-Support
- Noch vieles mehr

Neue Form-Elemente in Html5

- Eingaberange:

```
<input type="range" min="1" max="11" value="9" />
```

- Datumseingabe

```
<input type="date" />
```

- Zahleneingabe

```
<input type="number" min="1" max="10" value="1" />
```

Video Support in Html5

- Video mit Steuerelementen:

```
<video src=„tollesVideo.ogg“ width="320"  
      height="240" controls="controls" />
```

- Controls können per JavaScript / CSS angepasst werden.
- Lauffähig ohne Plugins

Composite Components (1)

Neue Komponenten in XHTML definieren

```
<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:cc="http://java.sun.com/jsf/composite/halloLib">
<h:head>
  <title>CC Demo</title>
</h:head>
<h:body>
  <h:form>
    <cc:hallo value="Andy Bosch" />
  </h:form>
</h:body>
</html>
```

Composite Components (2)

hallo.xhtml

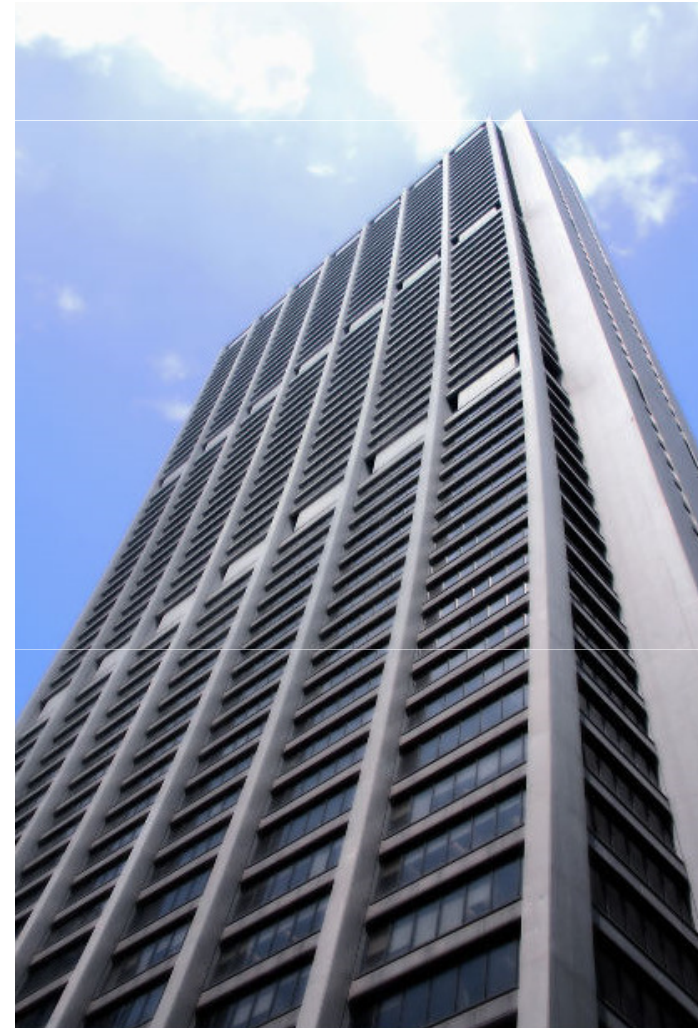
```
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:composite="http://java.sun.com/jsf/composite">
<body>
  <composite:interface>
    <composite:attribute name="value" required="true" />
  </composite:interface>
  <composite:implementation>
    <h:outputText
      value="Hallo #{cc.attrs.value}" />
  </composite:implementation>
</body>
</html>
```

Behauptung widerlegt



Behauptung 4: „Das Event-System ist mickrig“

- Diese Aussage ist auch 20 Jahre alt (und damit veraltet)
- Mit JSF 2.0 gibt es **SystemEvents**



System Events

- Werden bei speziellen Punkten im Lebenszyklus der Anwendung aktiviert, z.B.:
 - PostAddToViewEvent
 - PreRenderComponentEvent
 - PreRenderViewEvent
 - PostValidateViewEvent
- Bisherige Events waren auf Actions- oder ValueChanges beschränkt (naja, plus PhaseEvents)

System Events

```
<h:outputText id="beforeRenderTest1" >  
  <f:event type="preRenderComponent"  
    listener="#{SimpleEventController.processEvent}" />  
</h:outputText>
```

Es wird ein entsprechender Event an der Komponente registriert.

Listeners können prinzipiell auf drei Ebenen registriert werden:

- component: `UIComponent.subscribeToEvent()`
- view: `UIViewRoot.subscribeToEvent()`
- application: `Application.subscribeToEvent()`

Behauptung widerlegt



Behauptung 5: „JSF 2 und Portal geht ja noch überhaupt nicht“

- Auch wieder zu voreilig. Wohl nicht die aktuellen Blogs gelesen, oder? 😊



JSF 2.0 und Portlet 2.0

- Standard wird zur Zeit nicht anvisiert
 - Implementation first !
- Verschiedene Gruppen und Initiativen
 - Liferay → <http://www.portletfaces.org>
 - JBoss mit PortletBridge 3.0
 - Separater Trunk im MyFaces Repo

Behauptung widerlegt



Behauptung 6: „JSF ist nur ein Webframework von vielen Hunderten“

Auch hier muss ich widersprechen!

- **JSF ist DAS UI-Framework im JavaEE Stack**
- JSF integriert sich hervorragend mit anderen Technologien im Stack, z.B. JSR-299 (CDI), JSR-330 (Dependency Inj.), JSR-303 (BeanValidation) und anderen



JSR-299 / JSR-330

@Named

@SessionScoped

```
public class PersonBean implements Serializable {
```

```
    ...
```

@Inject

```
    private AccountChecker accountChecker;
```

```
    ... }
```

JSR-303 (Bean Validation)

```
@ManagedBean  
@SessionScoped  
public class ValidateBean {  
  
    @NotNull  
    private String firstname;  
  
    @Size (min=3)  
    private String lastname;  
  
    @Min (value=5)  
    private int alter;  
  
    ...  
}
```

Behauptung widerlegt



Behauptung 7:

„Die Entwicklung von JSF selbst ist langsam, weil es ein Standard ist“

- Ja, das stimmt gewissermaßen.
- Aber was macht ein Standard?
 - Er schaut, was sich im Markt etabliert.
 - Erfolgreiche Konzepte werden übernommen und standardisiert
 - „Best of breed“
- Langfristig ausgelegt
 - Stabilität und damit auch Investitionssicherheit

Jetzt aber mal ehrlich ...

- Das eben war ja schon sehr provokativ, oder?
→ Zugegeben, ja, aber inhaltlich nicht übertrieben oder falsch
- Wichtig ist zu verstehen, was die Ambitionen von JSF sind und wie die JSF Weiterentwicklung funktioniert.
- Lernen von anderen und offen sein für neue Ideen und Konzepte.
Auch JSF muss sich jeden Tag erneut beweisen !



Fazit

- JSF 2.0 stellt einen großen Schritt nach vorne dar
- JSF 2.0 ist ein mächtiges UI-Framework geworden
- JSF 2.0 ist **das** Standard UI-Framework im JavaEE Stack
- Die JSF-Community lebt nicht im Elfenbeinturm, sondern diskutiert aktiv Konzepte und Ideen anderer Frameworks und übernimmt diese ggf. in den Standard

Fragen? Kommentare? Meinungen?



Mehr über JSF unter
www.jsf-forum.de

Schulungen zu JSF?
andy.bosch@jsf-forum.de

Weiteres Fachsimpeln gerne am SENS-Stand

SENS