

Facelets

- der Durchbruch für JavaServer Faces?



Andy Bosch
IT-Berater und Projektleiter
Autor des JavaMagazins, JavaSpektrums und des ObjektSpektrums
SENS-Experte, www.softwareexperts.de
Betreiber von www.jsf-forum.de

Agenda

- Was ist JavaServer Faces?
- Wie funktioniert JavaServer Faces?
- Funktionsweise von Facelets



Agenda

- **Was ist JavaServer Faces?**
- Wie funktioniert JavaServer Faces?
- Funktionsweise von Facelets

Was ist JavaServer Faces?

- JavaServer Faces ist ein Framework !

JSF ist kein Produkt, sondern ein Standard / ein Rahmenwerk zur Erstellung von Webanwendungen.

- JavaServer Faces zielt auf das UI !

JSF bietet keine Unterstützung für ein O/R-Mapping, auch keine Unterstützung für Threading oder EJB-Unterstützung. JSF ist ein reines UI-Framework.



Agenda

- Was ist JavaServer Faces?
- **Wie funktioniert JavaServer Faces?**
- Funktionsweise von Facelets

Funktionsweise von JavaServer Faces

„Angriffspunkt“ für JSF-Requests ist immer das Faces-Servlet.
→ FrontController-Pattern

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app >

  <!-- ##### Faces Servlet ##### -->
  <servlet>
    <servlet-name>JavaServer Faces Servlet</servlet-name>
    <servlet-class>
      javax.faces.webapp.FacesServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <!-- ##### Mapping ##### -->
  <servlet-mapping>
    <servlet-name>JavaServer Faces Servlet</servlet-name>
    <url-pattern>/faces/*</url-pattern>
  </servlet-mapping>

</web-app>
```

Funktionsweise von JavaServer Faces

Für JSF-spezifische Themen existiert eine eigene Konfigurationsdatei: faces-config.xml

```
<faces-config>
  <navigation-rule>
    <from-view-id>/eingabe.jsp</from-view-id>
    <navigation-case>
      <from-outcome>success</from-outcome>
      <to-view-id>/ausgabe.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
  <navigation-rule>
    <from-view-id>/ausgabe.jsp</from-view-id>
    <navigation-case>
      <from-outcome>success</from-outcome>
      <to-view-id>/eingabe.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
  ...
</faces-config>
```

Backing Bean erstellen (1)

Ein erstes einfaches JavaBean

```
public class SquareBean {  
  
    private int length;  
  
    public int getArea() {  
        return length*length;  
    }  
  
    public int getLength() {  
        return length;  
    }  
  
    public void setLength(int i) {  
        length = i;  
    }  
  
}
```


Backing Bean erstellen (2)

Hinterlegen des JavaBeans in der Anwendungskonfigurationsdatei

```
<!DOCTYPE faces-config PUBLIC "-//Sun Microsystems,
  Inc.//DTD JavaServer Faces Config 1.0//EN"
  "http://java.sun.com/dtd/web-facesconfig_1_0.dtd">
<faces-config>

<managed-bean>
  <managed-bean-name>SquareBean</managed-bean-name>
  <managed-bean-class>
    com.edu.jsf.bean.SquareBean
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>

</faces-config>
```

→ JSF Bean-Management !!!

Funktionsweise von JavaServer Faces

Beispiel einer JSF-Seite

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
...
<f:view>
  <h:form>
    ...
    <td>Seitenlänge:</td>
    <td><h:inputText value="#{SquareBean.length}" /></td>
  </tr>
  <tr><td colspan="2" class="error">
    <h:messages />
  </td></tr>
  <tr><td colspan="1"><br />
    <h:commandButton action="success" value="Submit" />
  </td>
  ...
</h:form>
</f:view>
```

Agenda

- Was ist JavaServer Faces?
- Wie funktioniert JavaServer Faces?
- **Funktionsweise von Facelets**

JSF und JSF-nahe Projekte

In den letzten Monaten sind zahlreiche Projekte rund um JavaServer Faces entstanden. Ein kurzer Auszug:

- MyFaces
- **Facelets**
- Struts Shale
- ajax4jsf bzw. Ajax allgemein
- JBoss SEAM
- JSF-Spring
- ...

Facelets

Was ist Facelets?

- Subprojekt von JavaServer Faces
- Kopf dahinter: Jacob Hookom,
JSF-Expert Group Member
- Bietet Ergänzungen / Vereinfachungen für JSF
- Baut auf JSF auf (Referenzimplementierung oder MyFaces)

Funktionen im Überblick

- Templating (vergleichbar mit Tiles)
- Komposition von Komponenten
- Custom Logic Tags (if, repeat, ...)
- Designer-freundliche Seitenerstellung
- Einfache Möglichkeit, um Komponentenbibliotheken zu erzeugen
- Verbessertes Fehlerreporting

Facelets - Verwendung

Einbindung von Facelets in der faces-config.xml

```
<faces-config>

...
<application>
  <view-handler>
    com.sun.facelets.FaceletViewHandler
  </view-handler>
</application>

...
</faces-config>
```

Facelets – Eine erste Seite

```
<html xmlns='http://www.w3.org/1999/xhtml'  
      xmlns:h='http://java.sun.com/jsf/html'  
      xmlns:f='http://java.sun.com/jsf/core'>  
  
<head>  
  <title>Facelets-Test</title>  
</head>  
  
<body>  
  Guten Tag, #{PersonBean.firstname} #{PersonBean.nachname}  
</body>  
  
</html>
```

→ XHTML

→ Kann in jedem HTML-Editor bearbeitet werden

Facelets und JSF-Komponenten

```
<form id="myForm" jsfc="h:form">
  <input type="text" id="myInput"
    jsfc="h:inputText" value="#{SimpleBean.name}" />

  <input type="submit" value="Abschicken"
    id="myBtn" jsfc="h:commandButton" />

</form>
```

Ausgabe: `#{SimpleBean.name}`

Facelets und Navigation

```
<form id="myForm" jsfc="h:form">
  <input type="text" id="myInput"
    jsfc="h:inputText" value="#{SimpleBean.name}" />

  <input type="submit" value="Abschicken"
    action="success"
    id="myBtn" jsfc="h:commandButton" />

</form>
```

Ausgabe: #{SimpleBean.name}

Zusätzliche (unbekannte) Attribute werden von einem Editor und dem Browser meist ignoriert.

Facelets und Aktionsmethoden

```
<form id="myForm" jsfc="h:form">
  <input type="text" id="myInput"
    jsfc="h:inputText" value="#{SimpleBean.name}" />

  <input type="submit" value="Abschicken"
    action="#{SimpleBean.doSomething}"
    id="myBtn" jsfc="h:commandButton" />

</form>
```

Ausgabe: `#{SimpleBean.name}`

Pflege der Navigationsregeln in der faces-config.xml.

Facelets und Templating

```
<body>
<h1>
  <ui:insert name="title">Default Title</ui:insert>
</h1>
<p>
  <ui:insert name="body">Default Body</ui:insert>
</p>
</body>
```

„Template“

```
<body>
<ui:composition template="b2b-template.xhtml">

  <ui:define name="title">
    Dies ist der Titel
  </ui:define>

  <ui:define name="body">
    Dies ist der Body
  </ui:define>

</ui:composition>
</body>
```

„Template Client“

Facelets und Templating

```
<html>
<body>
<ui:composition template="/b2b-template.xhtml">

    Dummy-Text Dummy-Text

    <ui:define name="title">
        Dies ist der Titel
    </ui:define>

    Dummy-Text Dummy-Text

    <ui:define name="body">
        Dies ist der Body
    </ui:define>

    Dummy-Text

</ui:composition>
</body>
</html>
```

Wird ignoriert !

Vorteil:
Einzelne (Teil-)Seiten können zur Designzeit mit weiteren Elementen angereichert werden.

Facelets und Komponenten (1)

Definition einer Komponente in einer separaten Datei:

```
<ui:composition>

    <input type="button" value="Speichern" />
    <input type="button" value="Abbrechen" />
    <input type="button" value="Hilfe" />

</ui:composition>
```

Taglibrary erstellen:

```
<facelet-taglib>
    <namespace>http://www.jsf-forum.de/jsf</namespace>
    <tag>
        <tag-name>buttoncollection</tag-name>
        <source>component.xhtml</source>
    </tag>
</facelet-taglib>
```

Facelets und Komponenten (2)

Bekanntmachen der Bibliothek in der web.xml

```
<context-param>
  <param-name>facelets.LIBRARIES</param-name>
  <param-value>
    mytags.xml
  </param-value>
</context-param>
```

Verwendung des neuen Tags in einer Facelets-Seite:

```
<cu:buttoncollection />
```

→ Beachtung des DRY-Prinzips (Don't repeat yourself.)

Facelets und Komponenten (3)

Werteübergabe:

```
<cu:specialOutput wert="Hallo Welt" />
```

```
<ui:composition>
```

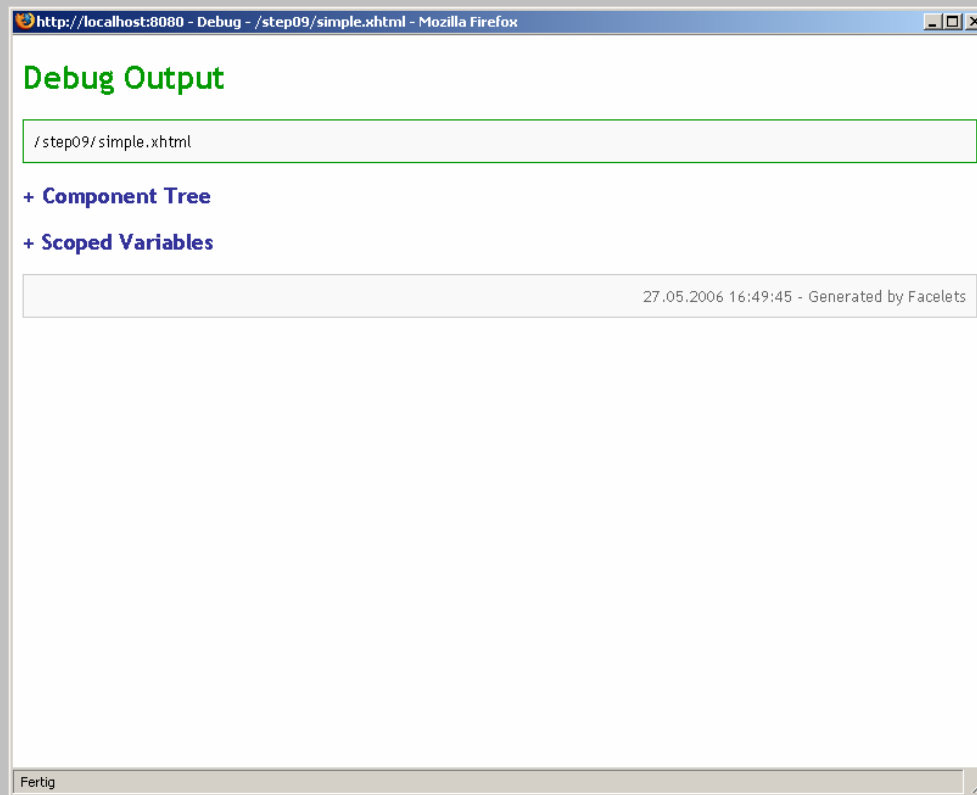
```
    Ausgabe: ${wert}
```

```
</ui:composition>
```


Facelets und Debugging

Einschalten des Debuggings:

```
<ui:debug hotkey="y" rendered="true" />
```



Fazit

- Facelets stellt einige sehr interessante Erweiterungen für JSF zur Verfügung.
- Der Tapestry-orientierte Ansatz ist für das Design von (Html-) Seiten durchaus sinnvoll.
- Die Templating-Funktionen funktionieren anstandslos.
- Schwierig ist momentan noch, eine lauffähige Umgebung aufzusetzen.
- Facelets lohnt sich, mittel- bis langfristig „im Auge zu behalten“

Links und Ressourcen

- <http://facelets.dev.java.net/>
Offizielle Webseite mit Download-Möglichkeit
- <http://facelets.dev.java.net/nonav/docs/dev/docbook.html>
Dokumentation (englisch)
- <http://www-128.ibm.com/developerworks/java/library/j-facelets/>
Tutorial und Einführung
- <http://www.jsf-central.com/>
Englischsprachiges JSF-Forum mit Verweisen auf Facelets-Tutorials
- <http://www.jsf-forum.de>
Deutschsprachiges JSF-Forum mit Facelets-Tutorials

Fragen ???



Weitere Tutorials und
viel Material unter:

www.jsf-forum.de

Gerne auch:
andy.bosch@jsf-forum.de