

Quo Vadis

JavaServer Faces



Andy Bosch
IT-Berater und Projektleiter
Buchautor
Autor des JavaSpektrums und des ObjektSpektrum
Betreiber von www.jsf-forum.de

Agenda

- Was ist JavaServer Faces?
- Wie funktioniert JavaServer Faces?
- Überblick über Tools, IDEs und sonstige Unterstützungen für Entwickler
- Status und Ausblick



Agenda

- **Was ist JavaServer Faces?**
- Wie funktioniert JavaServer Faces?
- Überblick über Tools, IDEs und sonstige Unterstützungen für Entwickler
- Status und Ausblick

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.

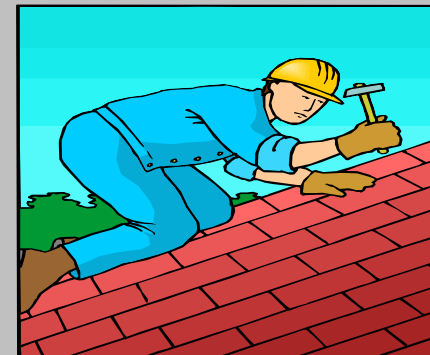


Was bietet JavaServer Faces?

- JSF ist ein Framework und liefert damit auch **Best-Practise** aus dem Bereich der Webentwicklung.
- JSF liefert erprobte und funktionierende Konzepte zur Umsetzung eines UI in Webanwendungen.

Beispiele für Konzepte in JSF sind:

- ✓ Model-View-Controller
- ✓ Navigationskonzept
- ✓ Bean-Management
- ✓ Eventhandling
- ✓ Datenvalidierung und -konvertierung
- ✓ ...



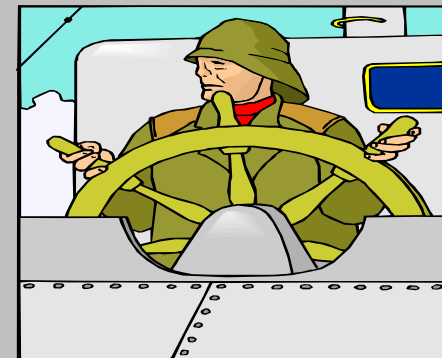
JavaServer Faces ist ein Standard



JSF ist ein erster offizieller Standard von Sun hinsichtlich der Erstellung von UIs von Webanwendungen.

JSF wurde im Rahmen des Java Community Process (JCP) entwickelt. Der zugrundeliegende JSR ist JSR 127.

Im Experten-Gremium sind viele namhafte Firmen wie IBM, BEA, IONA, Novell, Borland, HP, Oracle oder Siemens. Daher wird eine große Unterstützung seitens der Industrie erwartet.



Agenda

- Was ist JavaServer Faces?
- **Wie funktioniert JavaServer Faces?**
- Überblick über Tools, IDEs und sonstige Unterstützungen für Entwickler
- Status und Ausblick

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

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>Name:</td>
    <td><h:inputText value="#{Person.name}" /></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?
- **Überblick über Tools, IDEs und sonstige Unterstützungen für Entwickler**
- Status und Ausblick

Toolunterstützung

Eine breite Toolunterstützung konnte schon im Vorfeld angenommen werden, da viele „große“ Unternehmen im Expertengremium aktiv waren.

Heute bietet fast jede bekannte IDE eine JSF-Unterstützung:

- IBM im Rahmen des WSAD / Rational Web Developers
- Borland im Rahmen des Jbuilders
- Oracle mit Jdeveloper / ADF-Framework
- Sun im Rahmen des Creator Studios
- ...

Sun Creator Studio



- Guter graphischer Modeller
- Modeller jedoch nicht unbedingt für den Webdesigner geeignet
- Verschiedene Wizards zum Erstellen von Elementen
- Graphische Bearbeitung der Navigation
- Sämtliche Eigenschaften können über Properties-Dialoge verändert werden
- Preis: 99 \$

IBM Rational Web Developer



- Eigene JSF-Implementierung !
- Umfangreiche Wizards, Komponenten und Erweiterungen
- Durch proprietäre Erweiterungen nicht ohne weiteres auf anderen Application Servern ausführbar
- Konzepte für „Preactions“
- Guter graphischer Modeller
- Relativ teuer ☹

Oracle Jdeveloper mit ADF Faces

Basierend auf 10.1.2



- Keine eigene JSF-Implementierung, lediglich Erweiterungen!
- Graphischer Modeller (angekündigt für 10.1.3)
- Es können heute bereits die JSF-Bibliotheken miteingehängt werden.
→ Erweiterung der Komponentenpalette
- ADF Faces liefert zahlreiche weitere nützliche Komponenten
- Graphische Unterstützung der faces-config.xml
- Graphische Unterstützung der Navigationskonfiguration
- Graphischer Editor für sämtliche JSF-Eigenschaften

Weitere JSF-Unterstützung

- MyEclipseIDE
- NitroX von M7
- Exadel JSF-Studio
- zahlreiche freie Eclipse-Plugins
- ...

Agenda

- Was ist JavaServer Faces?
- Wie funktioniert JavaServer Faces?
- Überblick über Tools, IDEs und sonstige Unterstützungen für Entwickler
- **Status und Ausblick**

Ist JSF auch „praxistauglich“?

- Wichtig ist zunächst eine Unterscheidung zwischen der Spezifikation und einer Implementierung.
- Die Spezifikation ist gut, wenn auch kleinere Fragen noch ungeklärt sind.
- Es gibt mittlerweile viele Implementierungen, angefangen bei der Referenzimplementierung, über Open-Source (MyFaces) bis hin zu Implementierungen durch namhafte Hersteller (z.B. IBM)
- Insgesamt weisen alle Implementierungen bereits in den ersten Versionen eine sehr gute Stabilität und Performance vor.

Heutige Engpässe

- Vorinitialisieren von Beans („Preaction“) ist nur mittels Workarounds möglich.
- Standardkomponenten reichen nicht aus, es muss auf Dritt-Hersteller zurückgegriffen werden.
- Modeller für Oberflächen noch nicht ausgereift genug.
- „Garbage Collection“ von Managed Beans muss manuell gelöst werden.
- ...



Was bringt die Zukunft? Quo vadis ...?



- Spezifikation 1.2 wird als neuer **JSR 252** eingereicht
- Auszug aus dem JSR:

JavaServer Faces 1.2 provides a minimal enhancement of the JavaServer Faces 1.1 specification. It is not a new feature JSR. The **primary goal** of this JSR is handling the Faces side of the JSP/Faces alignment issues addressed in the parallel **JSR-245**.

Following is a list of **secondary goals** that will be addressed only if the primary goal is reached, and only if they can be addressed without delaying the schedule ...

Ziele des JSR-245



- JSR zielt auf JavaServer Pages 2.1
 - Kernaussage:
The purpose of JSP 2.1 is to improve alignment with JavaServer Faces and enhance ease of development.
- **Zusammenspiel von JSP und JSF soll verbessert werden !**
- It is a goal, therefore, of JSP 2.1 to enhance the expression language to meet the needs of JSF technology. Many of these enhancements are likely to be useful in other contexts as well.
- **Anpassung und Erweiterung der EL hinsichtlich Harmonie zwischen JSF und JSP.**

„Zweitrangige“ Ziele

- Einführung von XML-Schemas anstelle der DTDs für Config-Dateien
- Verbesserungen bei Verwendung von Frames und Fenstern
- Security-Verbesserungen bei Client Side State Saving
- Lösen des „Zweimal Button gedrückt“ – Problems
- Bugfixing hinsichtlich Portlet – Unterstützung

Konkrete Neuerungen



- JSF 1.2 ist KEIN New-Feature-Release
- Neue Expression Language (aus JSP-EL und JSF-EL wird Unified EL)
- Performance-Verbesserungen
- Aktueller Status (14.4.05): Public Review Draft

Ausblick auf JSF 2.0

- Verbesserte Toolunterstützung
- Zusätzliche Komponenten (z.B. für den Dateiupload)
- Client-Side Validation
- ...

→ Jedoch frühestens Ende 2005

Vielen Dank !!!



Bei Fragen:
andy.bosch@jsf-forum.de