



JavaServer Faces

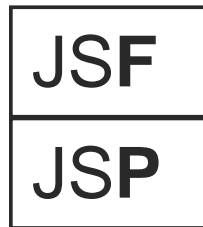
Urs Frei

[Inhalt]

- JSF Funktionsweise
 - Rückblick JSP
 - Bestandteile von JSF
- So einfach ist die Anwendung (Beispiel)
- Eclipse im Einsatz (Entwicklungsumgebung)
- Apache MyFaces

[Was ist JSF]

- Ein Framework für die Entwicklung von Benutzerschnittstellen in Java-Web-Applikationen
- Baut auf **JSP** auf



- Sun Spezifikation

[Erinnerungen an JSP]

- Nicht selten viel Java Code in der JSP
- Mühsamer Zugriff auf Javaobjekte
- Kommunikation mit Server nicht einfach

[Bsp. JSP]

```
<%@ page language="java" %>
<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">
<%fhsg.uebung8.AddressBean bean =
    (fhsg.uebung8.AddressBean)request.getAttribute("addressBean");
%>
<html><head><title>Übung 8 - Adressverwaltung</title></head>
<body bgcolor="#FFFFFF"><form>
  <h1>Adresse anzeigen:</h1>
  <p> Vorname:<%=bean.getFirstName() %></br>
    Nachname:<%=bean.getLastName() %></br>
    Adresse:<%=bean.getAddress() %></br>
    PLZ:<%=bean.getZip() %></br>
    Ort:<%=bean.getCity() %></br>
    Telefon:<%=bean.getPhone() %></br>
    E-Mail:<%=bean.getEmail() %></br>
    <a href="controller?action=showlist">zurück</a>
  </p>
</form></body></html>
```

JSF soll Arbeitserleichterung für Entwickler bieten, in:

- Komponenten
 - Web Applikation wird zusammengebaut
 - Möglich eigene Komponenten bauen
- Datentransfer
 - Konverter
 - Validierung
- Zustandsspeicherung
 - Speicherung Applikationszustand
- Ereignisbehandlung
 - Server kann Clientereignisse behandeln

[Bestandteile für JSF]

- web.xml
- JSF Files
- faces-config.xml
- Java Klassen in der Java Bean Form
- Expression Language (EL)

[JSF File]

- Ist ein *.jsp mit taglib
- Was sind taglib?
 - Ermöglicht die Definition eigener Tags
 - Verwendung:
 - „import“ Definition
 - Verwenden wie die üblichen Tags

[Beispiel taglib Verwendung]

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
  <head>
    <title>Gespeichert</title>
  </head>
  <body>
    <f:view>
      <h:form>
        <h:outputLabel>Die Daten sind gespeichert:</h:outputLabel>
        <br>
      </h:form>
    </f:view>
  </body>
</html>
```

[faces-config.xml]

- Managed-Beans
 - Zugriff auf die Daten im Modell
- Navigationsregeln
 - Welche Seite folgt auf welche Seite
- Renderer
 - Wie sollen sich die Komponenten zeichnen
- Validatoren
 - Welche Eingabe vom User ist gültig
- Konverter
 - Wie sollen die Eingabedaten konvertiert werden

Managed-Beans (faces-config.xml)

- Klassen, die nach den Konventionen von Java Bean erstellt wurden
 - Member mit set und get
- Werden im faces-config.xml definiert
 - Sehr einfacher Zugriff vom JSP

[Expression Language (EL)]

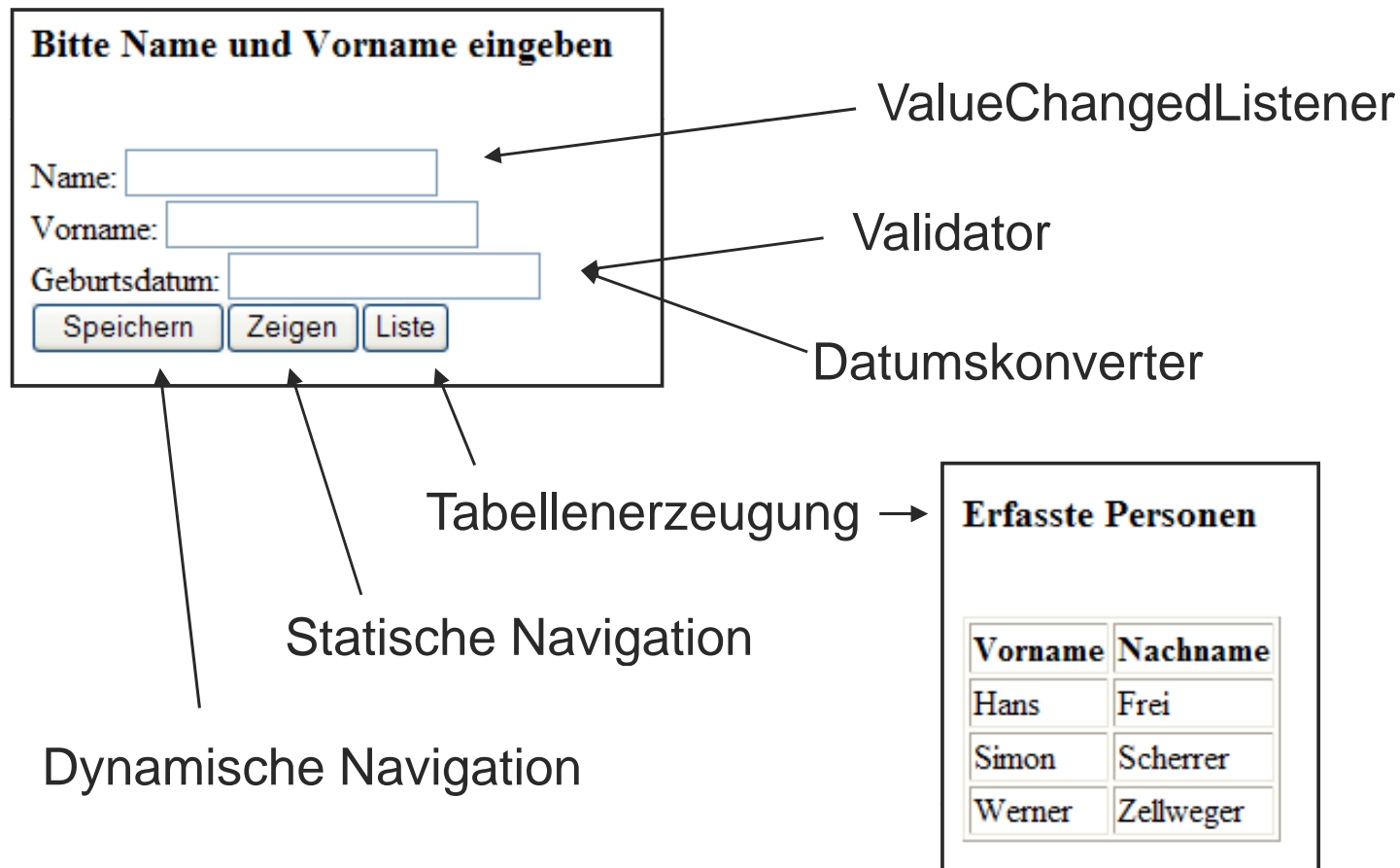
- Ermöglicht den Zugriff auf die Managed-Beans aus dem JSP
- Aufbau:
„#{managedBeanName.MemberName}“

[Voraussetzungen für JSF]

- Entwicklungsumgebung Eclipse
 - WTP (Eclipse Project) für JSF 1.2 neuste Version nötig (I20070313)
 - Exadel Studio, Plugin für Eclipse (vom hören sagen)
- Java EE 5 mit Sun Application Server PE 9
- Tomcat 5.5 keine Unterstützung, da alte JSP Version

[So einfach ist die Umsetzung!!]

■ Beispiel „Anwendung“:



[Web.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>
      javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.faces</url-pattern>
  </servlet-mapping>
</web-app>
```

[Input.jsp]

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>...
<body><f:view>
  <h3>Bitte Name und Vorname eingeben</h3>
  <br>
  ...
  Vorname:
  <h:inputText value="#{person.firstName}"/>
  <br>
  <h:commandButton value="Zeigen" action="show"/>
  ...
</html>
```


[Managed-Bean (faces- config.xml)]

```
<managed-bean>  
  <managed-bean-name>person  
</managed-bean-name>  
  <managed-bean-class>  
    ch.naturasoft.jsf.address.model.Person  
</managed-bean-class>  
  <managed-bean-scope>session  
</managed-bean-scope>  
</managed-bean>
```

[Navigationsregeln]

- Zwei Arten
 - Dynamisch Navigation
 - Statisch Navigation

[Dynamische Navigation]

- JSP (Input.jsp)
 - `<h:commandButton value="Speichern" action="#{saver.saveData}"/>`
- Managed-Bean (java)
 - `public String saveData(){
 return "ok";}`
- faces-config.xml
 - `<navigation-rule>
 <from-view-id>/Input.jsp</from-view-id>
 <navigation-case>
 <from-outcome>ok</from-outcome>
 <to-view-id>/Saved.jsp</to-view-id>
 </navigation-case>
</navigation-rule>`

[Statische Navigation]

- JSP (Input.jsp)
 - `<h:commandButton value="Zeigen" action="show"/>`
- faces-config.xml
 - `<navigation-rule>`
 - `<from-view-id>/Input.jsp</from-view-id>`
 - `<navigation-case>`
 - `<from-outcome>show`
 - `</from-outcome>`
 - `<to-view-id>/Show.jsp</to-view-id>`
 - `</navigation-case>`
 - `</navigation-rule>`

[ValueChangeListener]

■ JSP (Input.jsp)

- `<h:inputText value="#{person.lastName}">`
 `<f:valueChangeListener`
 `type="ch...NameValueChangeListener"/>`
`</h:inputText>`

■ Java

- ```
public class NameValueChangeListener implements ValueChangeListener
{
 public void processValueChange(ValueChangeEvent arg0) throws
 AbortProcessingException {
 ...
 }
}
```

# [ Validator ]

- JSP (Input.jsp)

```
<h:inputText value="#{person.birthday}">
 <f:validator binding="#{dateValidator}"/>
</h:inputText>
```

- Java

```
public class DateValidator implements Validator {
 public void validate(FacesContext f, UIComponent ui, Object o)throws
 ValidatorException {
 ...
 }
}
```

- faces-config.xml

```
<managed-bean>
 <managed-bean-name>dateValidator</managed-bean-name>
 <managed-bean-class>
 ch...DateValidator</managed-bean-class>
 <managed-bean-scope>none</managed-bean-scope>
</managed-bean>
```

# [ Konverter ]

- Vorgefertigter Konverter für Datum
- JSP (Input.jsp)  

```
<h:inputText value="#{person.birthday}">
 <f:convertDateTime/>
</h:inputText>
```
- Erzeugung eigener Konverter möglich

# [ Tabellenerzeugung ]

- JSP (List.jsp)

```
<h:dataTable var="tmpPerson" value="#{model.persons}" border="1">
 <h:column>
 <f:facet name="header">
 <h:outputText value="Vorname"/>
 </f:facet>
 <h:outputText value="#{tmpPerson.firstName}"/>
 </h:column>
 <h:column>
 <f:facet name="header">
 <h:outputText value="Nachname"/>
 </f:facet>
 <h:outputText value="#{tmpPerson.lastName}"/>
 </h:column>
</h:dataTable>
```

- **Voraussetzungen: alle Manages-Bean sind registriert (config-faces.xml)**



# Eclipse im Einsatz

## Bitte Name und Vorname eingeben

Name:

Vorname:

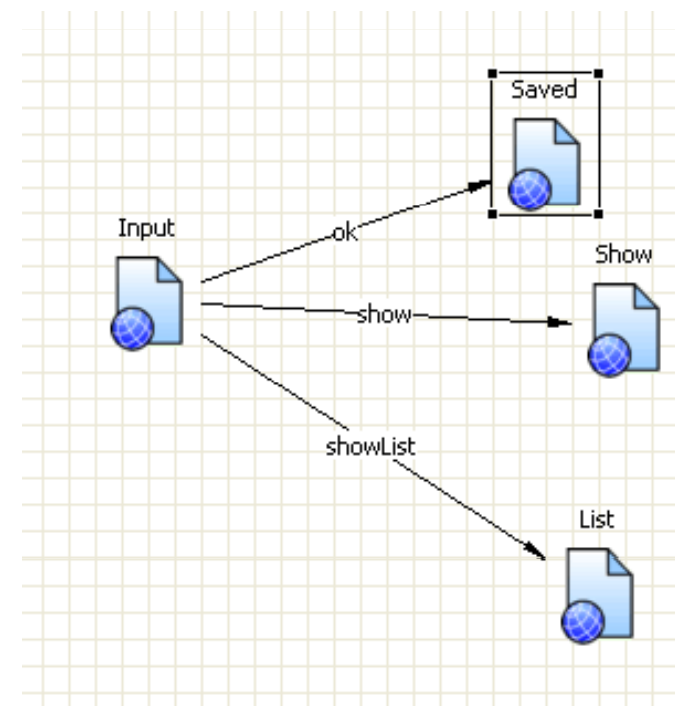
Geburtsdatum:

messages

```
<?taglib uri="http://java.sun.com/jsf/html" prefix="h" ?>
<?taglib uri="http://java.sun.com/jsf/core" prefix="f" ?>
<html>
<head>
 <title>Personendaten eingeben</title>
</head>
<body>
 <f:view>
 <h3>Bitte Name und Vorname eingeben</h3>

 <h:form id="inputForm">
 Name:
 <h:inputText value="#{person.lastName}">
 <f:valueChangeListener type="ch.naturasoft.j
 </h:inputText>


```



# [ MyFaces (Apache) ]

---

- Implementation des JSF Standards
- Mehr Komponenten
- Noch weitere Vereinfachungen
- Hat div. Unterprojekte mit verschiedenen Weiterentwicklungen

# [ Beispielkomponenten ]

## dataTable Example

⏪ ⏩ 1 2 3 4 5 6 7 8 9 ⏪ ⏩

994 Autos gefunden, zeigend 10 Autos, von 11 bis 20. Seite 2 / 100

	Modell	Farbe
11	Car Type 11	blue
12	Car Type 12	blue
13	Car Type 13	blue
14	Car Type 14	blue
15	Car Type 15	blue
16	Car Type 16	blue
17	Car Type 17	blue
18	Car Type 18	blue
19	Car Type 19	blue
20	Car Type 20	blue

⏪ ⏩ 1 2 3 4 5 6 7 8 9 ⏪ ⏩

994 Autos gefunden, zeigend 10 Autos, von 11 bis 20. Seite 2 / 100

client-side tab switching

Tab1 Tab2 Tab3

A common paragraph

A checkbox

This Tab is active!

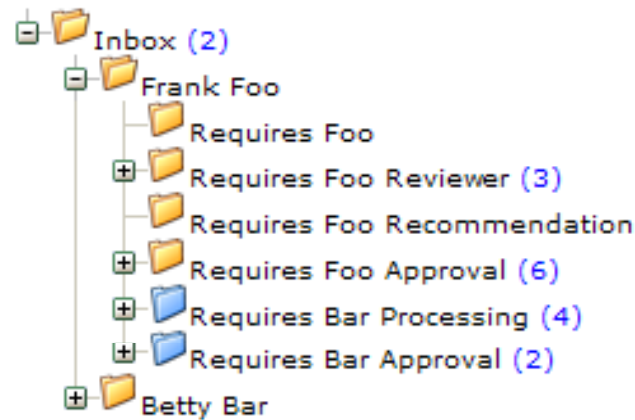
---

Tab 1 visible  
 Tab 2 visible  
 Tab 3 visible

Common submit button

# [ Beispielkomponenten ]

## Tree2 w/server-side toggle



## Editable Content With Default Toolbar

< März 2007 >

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

add sample entries

add sample holiday

Mar 12, 2007

8 <sup>00</sup>	
9 <sup>00</sup>	
10 <sup>00</sup>	
11 <sup>00</sup>	

# [ Ausblick: ]

---

- Web Applikationen erstellen mit
  - Apache MyFaces
  - Ajax
  - Spring (Hibernate)
- Ein Bsp. dazu findet Ihr im Buch

# [ Links ]

---

- Eclipse [www.eclipse.org](http://www.eclipse.org)
- WTP <http://www.eclipse.org/webtools/>
- Exadel-Studio [www.exadel.com/](http://www.exadel.com/) Plugin für Eclipse
- Java EE 5 [java.sun.com](http://java.sun.com)

# [ Referenzen ]

---

- JSF@Work ISBN3-89864-401-4
- <http://java.sun.com/javaee/javaserverfaces/> JSF Spezifikation
- <http://www.jsf-forum.de/> mit Tutorials
- <http://myfaces.apache.org/> Apache myFaces
- <http://www.irian.at/myfaces.jsf>