

Struts

Michael Hartl, Laurent Steurer
06.06.2007

Inhaltsangabe

- ◉ Einführung / Geschichte
- ◉ Motivation
- ◉ Beispiel
- ◉ Architektur / Komponenten
- ◉ Konfiguration
- ◉ Fazit

Einführung

- ◉ Open Source Framework für Web-Anwendungen
- ◉ Kompatibel zur J2EE Plattform
- ◉ Teil des umfangreichen Apache Jakarta Projekts
- ◉ Struts: bautechnischer Ausdruck (engl.) für „Verstrebungen, Abstützungen“
- ◉ basiert auf Standard Java APIs
- ◉ basiert auf Model-View-Controller-Pattern
- ◉ basiert auf JSP- und Servlet-Technologie
- ◉ aktuelle Version ist Struts 1.3

Geschichte

● 2000:

- Craig R. McClellan
 - war auch federführend bei der Entwicklung von Java-Server-Faces
- Entwicklung eines MVC Frameworks für Java-Plattform

● Januar 2001:

- Apache Software Foundation veröffentlicht MVC Framework: Struts

Motivation - Warum noch ein Framework?

- ◉ Bewährte Architekturen und Entwurfsmuster können benutzt werden, anstatt sie immer wieder neu zu implementieren
- ◉ Erleichterung von Standardaufgaben in Web-Anwendungen:
 - > Formularbehandlung
 - > Mehrsprachigkeit
- ◉ Schnellere Entwicklungszeiten
- ◉ Statt unschönem HTML-Code (hoffentlich) schönere JSP CustomTags

Struts Übersicht

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ MVC Architektur (Model 2)
 - > Model: z.B.: JavaBeans, EJBs, Datenbank, EIS, ...
 - > View: JSP
 - > Controller: Teile von Struts und eigene Controller Klassen
- ◉ Mehrsprachigkeit
- ◉ JSP Taglibs
- ◉ Server- und clientseitige Eingabevalidierung möglich
- ◉ XML basierte Konfiguration für die Formularnavigation
- ◉ Tiles Framework zur Steuerung des Layouts
- ◉ Deklaratives Exception Handling

Architektur

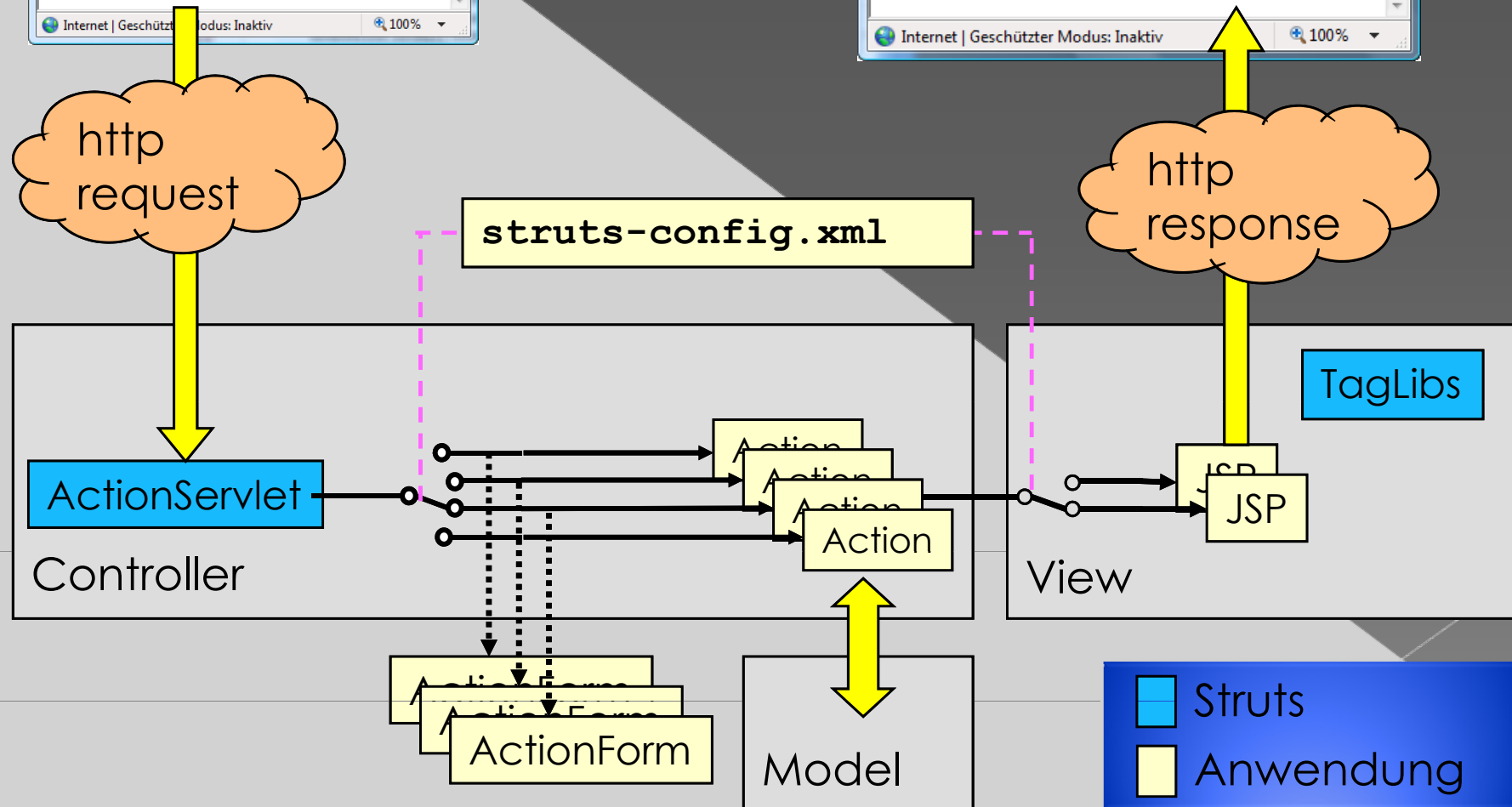
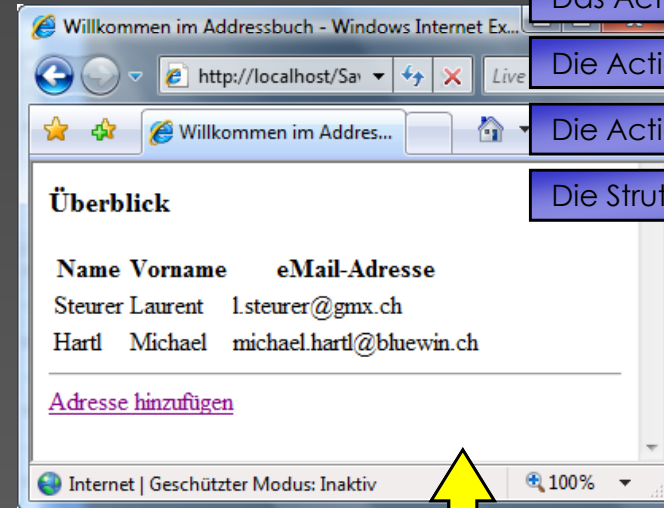
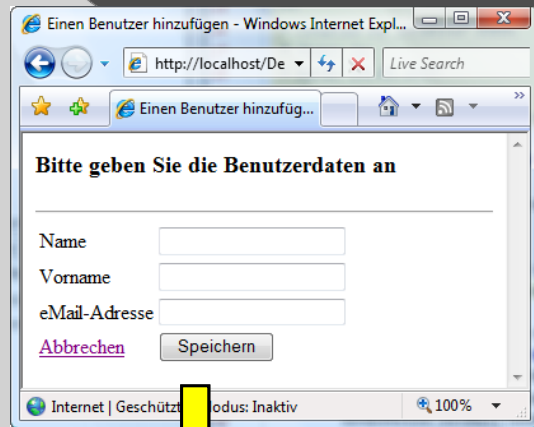
Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration



Action Servlet

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ Ablaufsteuerung
- ◉ Zentrale Controller-Instanz der Struts Web-Anwendung
- ◉ Wird über ein Konfigurationsfile (struts-config.xml) gesteuert
- ◉ Nimmt Parameter aus den Formularen entgegen und füllt damit die **ActionForm** Objekte
- ◉ Leitet die **ActionForm** Objekte nach erfolgreicher Validierung an die **Action** Objekte weiter

ActionForm

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ Erlauben eine einfache Übernahme der Formulardaten in die Properties einer JavaBean Klasse
- ◉ Vorbelegung von Formularfeldern mit Werten aus den JavaBean Properties
- ◉ Jedes HTML-Eingabefeld besitzt ein gleichnamiges Bean-Property im ActionForm
- ◉ Optional können die übernommenen Eingabewerte in einer **validate()** Methode überprüft werden
- ◉ Werden abgeleitet von **org.apache.struts.action.ActionForm**

Properties

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

Einen Benutzer hinzufügen - Windows Internet Expl...

http://localhost/De Live Search

Einen Benutzer hinzufü...

Bitte geben Sie die Benutzerdaten an

Name

Vorname

eMail-Adresse

[Abbrechen](#)

Internet | Geschützter Modus: Inaktiv 100%

Verbindung durch die Namensgleichheit des JSP Tag Attribut-Werts und des JavaBean Property-Namens

ActionForm

AddressForm

name
firstName
validate()

```
<html:form action="/SaveAddress">
<tr>
  <td><bean:message key="address.name"/></td>
  <td><html:text property="name"/></td>
</tr>
...
<html:submit property="submit">
  <bean:message key="detail.save"/>
</html:submit>
...
</html:form>
```

addressDetails.jsp

Verbindung zwischen der Aktion **SaveAddress** und der Klasse **AddressForm** laut struts-config.xml

Validierung

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ Neben der **validate()** Methode in der **ActionForm** Klasse kann die Validierung auch deklarativ über eine XML Datei erfolgen.
 - > Dazu beinhaltet Struts eine Reihe von Standard-Validatoren (z.B. für Beträge, Zahlen, e-mail Adresse, ...) und stellt dazu Standard-Fehlermeldungen bereit.
 - > Einer dieser Validatoren ("mask") kann über Regular Expressions parametrisiert werden. Damit lassen sich die meisten Einzelfeldprüfungen realisieren.
- ◉ Mit einem Eingabefeld können mehrere Validatoren verknüpft werden.

Fehler-Behandlung

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

```
public class AddressForm extends ActionForm
{
    private String name;
    private String firstName;
    ...
    ... Getter und Setter Methoden
    ...
    public ActionErrors validate(ActionMapping mapping,
                                   HttpServletRequest request)
    {
        ActionErrors errors = new ActionErrors();
        if ((name == null) || (name.equals("")))
        {
            errors.add("name", new ActionMessage("detail.error.name.required"));
        }
        return errors;
    }
}
```

Implementierung der **validate()**
Methode ist optional

Ein leeres **errors** Object steht für
eine erfolgreiche Validierung

Textschlüssel

AddressForm.java

Fehlerbehandlung II

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

Bitte geben Sie die Benutzerdaten an

- Bitte geben Sie einen Namen ein !
- eMail - Address enthält kein @ !

Name

Vorname

eMail-Adresse

[Abbrechen](#)

Speichern

Das `<html:errors>` Tag benötigt diese beiden Textschlüssel

```
errors.header = Eingabefehler  
errors.footer = <hr>
```

```
detail.error.name.required=Bitte geben Sie einen Namen ein !  
detail.error.email.noat=eMail - Address enthält kein @ !
```

```
...  
<BODY>  
  
<html:errors />  
  
<html:form....
```

AddressDetails.jsp

Mit diesem Tag werden alle Fehlermeldungen im **ActionErrors** Object ausgegeben

MessageResource.properties

TagLibs

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

Erweiterung der Funktionalität durch Tag-Bibliotheken

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
```

- ◉ Bean-Tags
 - > Zugriff auf Beans und deren Properties (get/set)
 - > <jsp:useBean id=„mybean“ class=„myBean“/>
- ◉ HTML-Tags
 - > Form Tags (z.B. <html:form>, <html:text>)
 - > Tag Attribute für JavaScript (z.B. Event Handling)
 - > Mehrsprachigkeit
- ◉ Logic-Tags
 - > Bedingte Textausgabe
 - > Iteration über Objekt-Collections (<logic:iterate>)
 - > Anwendungsfluss-Management
 - > Existenzprüfung (<logic:present>)
- ◉ JSTL JSP Standard Tag Library
 - > Iterationen, Kontrolle, Import via ftp oder http, XML, Datenbanken ...

Action Klasse

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ Sind Teil des Contollers und stellen die Verbindung zum Model her
- ◉ Werden abgeleitet von **org.apache.struts.action.Action**
- ◉ Bearbeiten den Request bzw. die Daten aus dem ActionForm Objekt
- ◉ Erzeugen eine Ausgabe über den Response oder leiten zu einer Ausgabeseite weiter
- ◉ müssen Thread-Safe sein, da das Struts Framework nur eine Instanz pro Klasse hält
- ◉ Das gewünschte Verhalten wird durch Überschreiben der **execute()** Methode implementiert

Action Klasse II

execute()-Methode

„Business Logik“ Zeile

```
public ActionForward execute(ActionMapping, ActionForm,  
    HttpServletRequest, HttpServletResponse)  
{  
    ...  
    // Zugriff auf Session und Adress-Liste ablegen  
    HttpSession session = httpRequest.getSession();  
    List<Address> addressList = (List<Address>)session.getAttribute("list");  
    ...  
  
    // Umleiten auf entsprechende JSP-Seite  
    // (in struts-config.xml konfiguriert)  
    return actionMapping.findForward("continue");  
}
```

AddressAction.java

Mapping Namen für
die Folgeseite

Struts-Konfiguration

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

- ◉ Identifikation der Elemente einer Struts Anwendung erfolgt über logische Namen (Mapping)
 - > Zuordnung logischer Namen zu Java Server Page, Servlet, Klasse
 - > Start der Anwendung: Einlesen erzeugt Mapping-Tabelle

Struts-Konfiguration II

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

```
...
<servlet>
  <servlet-name>action</servlet-name>
  <servlet-class>
    org.apache.struts.action.ActionServlet
  </servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
</servlet>

<servlet-mapping>
  <servlet-name>action</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
...
```

Struts Konfigurationsdatei

web.xml

Struts-Konfiguration III

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

```
<struts-config>
```

```
<form-beans>
```

```
<form-bean
```

```
  name="addressForm"
```

```
  type="adressbook.AddressForm"/>
```

```
</form-beans>
```

```
<global-forwards>
```

```
  <forward name="initList" path="/List.do"/>
```

```
</global-forwards>
```

```
<action-mappings>
```

```
  ... (siehe nächste Folie)
```

```
</action-mappings>
```

```
</struts-config>
```

Zuordnung von Formularnamen zu ActionForm Klassen (diese Namen werden im Action-Mapping benötigt)

Anwendungs-globales Mapping von Namen auf Seiten

Konfiguration einer Aktion

struts-config.xml

Struts-Konfiguration IV

Action Mapping

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

Mapping Name laut
web.xml URL-Mapping

Klassenname der
Action-Subklasse

Name der vorher
definierten ActionForm

```
<action-mappings>
  <action path="/SaveAddress"
    type="addressbook.AddressAction"
    name="addressForm"
    input="/addressDetail.jsp"
    scope="request"
    validate="true">

    <forward name="continue"
      path="/List.do" />

  </action>
</action-mappings>
```

struts-config.xml

validate()
in der
ActionForm
aufrufen

In der Action Klasse lokal gültige
Namen für Weiterleitungen

Struts-Konfiguration V

Mehrsprachigkeit

Struts Übersicht

Das Action Servlet

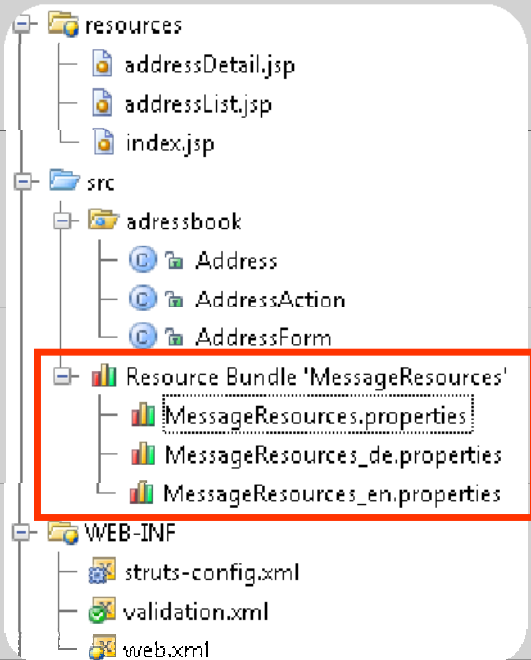
Die Action Form Klassen

Die Action Klassen

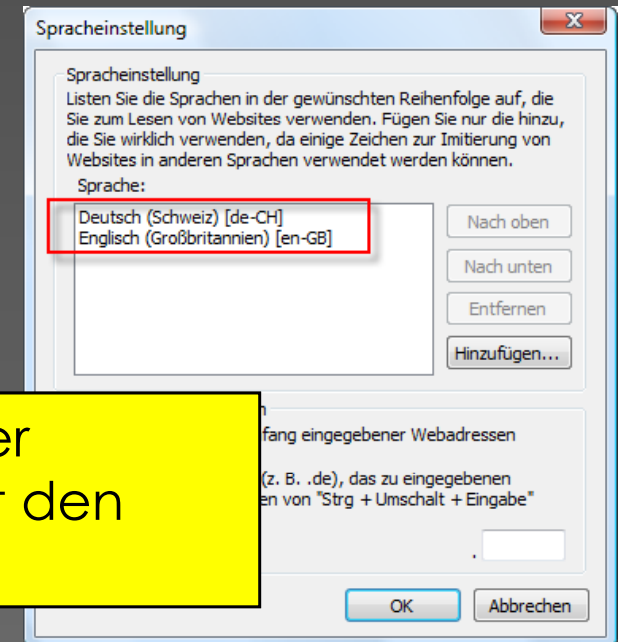
Die Struts Konfiguration

```
<message-resources parameter="MessageResource" />
```

struts-config.xml



Name (Präfix) der Property-Files mit den Text-Ressourcen



Textschlüssel für die Struts-Tags

```
list.title=Willkommen im Addressbuch
```

MessageResource_de.properties

```
<bean:message key="list.title" />
```

AddressDetails.jsp

Struts-Konfiguration V

Tiles

Struts Übersicht

Das Action Servlet

Die Action Form Klassen

Die Action Klassen

Die Struts Konfiguration

```
<tiles:insert>  
<tiles:insert>  
<tiles:insert>
```

layout.jsp

```
<definition name="templateDefinition"  
    path="/layout.jsp">  
    <put name="title"  
        value="This is the title." />  
    <put name="header"  
        value="header.jsp" />  
    <put name="body"  
        value="body.jsp" />  
    <put name="footer"  
        value="footer.jsp" />  
</definition>
```

struts-config.xml

header.jsp

body.jsp

footer.jsp

Fazit

○ Vorteile:

- Komplexe Webanwendung wird in kleine Komponenten aufgeteilt (Bottom Up)
- Anwendung leicht erweiterbar und Komponenten durch Trennung wiederverwendbar
- Für grosse Anwendungen geeignet
 - Bei Wachstum der Anwendung bleiben Überblick und Variabilität erhalten

○ Nachteile:

- Aufteilung der Anwendung in Komponenten
→ hohe Komplexität
- Noch in Weiterentwicklung und Veränderung
→ Änderungen im Code möglicherweise nötig
- Das Erstellen der Konfigurationsdatei ist sehr fehlerträchtig
→ die XML Attributnamen sind oft wenig intuitiv
- Action-Klasse muss Thread-Safe sein (jeweils nur eine Instanz)
- Einführung von Struts erfordert Refactoring der Applikation

Quellen

- ◉ Struts Home

<http://jakarta.apache.org/struts/>

- ◉ Wikipedia

<http://de.wikipedia.org/wiki/Struts>

- ◉ Buch Java-EE 5

Addion-Wesley

ISBN: 3-8273-2362-2