9.6 JSP und Tags

- Problem:
  JSP sind Mischung aus XML Fragmenten und Kode

$\Rightarrow$ schlechte Lesbarkeit

$\Rightarrow$ Java Kenntnisse erforderlich

- Lösung: Tags—Programmiermuster in XML Syntax

9.6.1 Tag Dateien

- Tag Definition durch *Tag Datei*
- ab JSP Version 2.0 (früher ziemlich umständlich)
- Variante einer JSP
- Dateiendung `.tag` erforderlich
- `JspContext jspContext` anstelle von `pageContext`
- Beispiel: Abstraktion der Additions-JSP

Beispiel: Datei `wrap.tag`

```
<%-- Eine Tag Definitionsdatei --%>
<%@ tag %>
<%-- Viele Attribute, ähnlich page --%>
<%@ attribute name="title" required="true" %>
<%-- Notwendiges Attribut für das Tag --%>
<html>
  <head><title>${title}</title></head>
  <body>
    <jsp:doBody/>
    <%-- Hier wird der Rumpf des Tag eingefügt --%>
  </body>
</html>
```

Verwendung des <wrap> Tags

```
<%@ taglib prefix="foo" tagdir="/WEB-INF/tags" %>
<%-- Assoziation des Präfix mit Verzeichnis --%>
<foo:wrap title="Addition">
  The sum of ${param.x} and ${param.y} is ${param.x+param.y}
</foo:wrap>
```

Weitere Tag Features

- Kopieren des Rumpfes in Variable ⟨*name*⟩

  ```
  <jsp:doBody var="⟨name⟩"/>
  ```

- Beispiel: `image.tag`

  ```
  <%@ tag %>
  <jsp:doBody var="src"/>
  <img src="http://www.brics.dk/ixwt/images/${src}"/>
  ```

- Verwendung

  ```
  <%@ taglib prefix="foo" tagdir="/WEB-INF/tags" %>
  <foo:image>widget.jpg</foo:image>
  ```

Kommunikation zwischen Tag und JSP: date.tag

```
<%@ tag import="java.util.*" %>
<%@ variable name-given="date" %>
<%@ variable name-given="month" %>
<%@ variable name-given="year" %>
<%
   Calendar cal = new GregorianCalendar();
   int date = cal.get(Calendar.DATE);
   int month = cal.get(Calendar.MONTH)+1;
   int year = cal.get(Calendar.YEAR);
   jspContext.setAttribute("date", String.valueOf(date));
   jspContext.setAttribute("month", String.valueOf(month));
   jspContext.setAttribute("year", String.valueOf(year));
%>
<jsp:doBody/>
```

Verwendung von <date>

```
<%@ taglib prefix="foo" tagdir="/WEB-INF/tags" %>
<foo:date>
  In the US today is ${month}/${date}/${year},
  but in Europe it is ${date}/${month}/${year}.
</foo:date>
```

9.6.2 Domain-Specific Language

- Tag-Bibliothek kann eine anwendungsspezifische Sprache (Domain-Specific Language, DSL) definieren

$\Rightarrow$ konziser Anwendungskode

- Trennung zwischen Anwendungslogik und grafischer Gestaltung

Beispiel

```
<%@ taglib prefix="poll" tagdir="/WEB-INF/tags/poll" %>
<poll:quickpoll title="Quickies" duration="3600">
  <poll:setup>
    The question has been set to "${question}".
  </poll:setup>
  <poll:ask>
    ${question}?
    <select name="vote">
      <option value="yes">yes</option>
      <option value="no">no</option>
    </select>
    <input type="submit" value="vote"/>
  </poll:ask>
  <poll:vote>You have voted ${vote}.</poll:vote>
```

```
  <poll:results>
    In favor: ${yes}<br/>
    Against: ${no}<br/>
    Total: ${total}
  </poll:results>
  <poll:timeout> Sorry, the polls have closed.</poll:timeout>
</poll:quickpoll>
```

Beispiel: quickpoll.tag

```
<%@ attribute name="title" required="true" %>
<%@ attribute name="duration" required="true" %>
<%@ attribute name="sheet" %>
<% response.addDateHeader("Expires", 0);
   application.setAttribute("duration",
                    jspContext.findAttribute("duration")); %>
<html>
  <head><title>${title}</title>
    <% if (jspContext.findAttribute("sheet")!=null) { %>
    <link rel="stylesheet" href="${sheet}" type="text/css">
    <% } %>
  </head>
  <body><jsp:doBody/></body>
</html>
```

Eigenschaften des DSL Ansatzes

- Veränderung der Gestaltung der Anwendung durch Änderung des Inhalts von `<poll:quickpoll>`, ohne Änderung der Programmlogik in den Tags
- Impliziter Vertrag zwischen Tag-Programmierer und Anwendungsgestalter
  - Äußeres Tag `<quickpoll>`, darin `<question>`, `<ask>`, `<vote>`, `<results>` und `<timeout>` *in dieser Reihenfolge*
  - Tag-Programmierer definiert die Attribute `question`, `vote`, `yes`, `no`, and `total`
  - Anwendungsgestalter sendet ein `vote`-Feld innerhalb des `<ask>` Tags
- Vertrag wird nicht überprüft :-(

9.6.3 Tag Bibliotheken und die JSTL

- Eine Tag Bibliothek (*tag library*) ist eine Sammlung von Tags für einen Anwendungsbereich
- Ersatz für direkte Programmierung
- Beispiele
  - Jakarta Taglibs: Caching, Datum und Zeit, Datenbankzugriff, Unterstützung für Internationalisierung, Formularunterstützung und -validierung, IO-Protokolle, Logging, Mail, reguläre Ausdrücke, HTML Extraktion, uvam
  - JSP Standard Tag Library (JSTL, Teil des JSP Standards): Variable, Ausgabe, Exceptions, If-then-else, Iterationen, URL, Stringformatierung, SQL-Anfragen, XML-Bearbeitung

Beispiel: Visitenkartenserver, Startseite

```
<html>
  <head>
   <title>Search For Business Cards</title>
  </head>
  <body>
    <form method="GET" action="select_card.jsp"
      >Enter a search string:
      <input type="text" name="pattern"/>
      <br/>
      <input type="submit" value="Go"/>
    </form>
  </body>
</html>
```

Beispiel: Visitenkartenserver, Selektion

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x"%>
<c:import url="cardlist.xml" var="xml"/>
<x:parse xml="${xml}" var="cardlist" scope="session"/>
```

```
<html>
  <head><title>Select A Card</title></head>
  <body>
    <ul>
      <c:set var="pattern" value="${param.pattern}"/>
      <x:forEach varStatus="i"
                 select="$cardlist//*[local-name()='card']">
        <x:if select="*[contains(.,$pattern)]">
          <li><a href="display_card.jsp?i=${i.index}">
                <x:out select="*[local-name()='name']/text()"/>
              </a></li>
        </x:if>
      </x:forEach>
    </ul>
  </body>
</html>
```

9.6.4 Zusammenfassung JSP

- Alternative zur direkten Java Servlet Programmierung
- XML Syntax mit Java Kodefragmenten
- Übersetzt in Servlets
- Tag Bibliotheken können anwendungsspezifische Programmmuster durch XML Elemente ausdrücken
- Problem: Fehlersuche
  - Fehlermeldungen während/nach Übersetzung in Servlet
  - Deployment