

Java Server Pages



Christoph Süsens & Matthias Holdorf

Definition JSP

- Eine auf JHTML entwickelte Webprogrammiersprache
- Dient zur einfachen Erzeugung dynamischer HTML & XML Ausgaben mittels Webserver
- Erlaubt Java-Code & spezielle JSP-Aktionen in statischen Inhalt einzubetten
- Vorteil Logik unabhängig vom Design (MVC)

Statisch

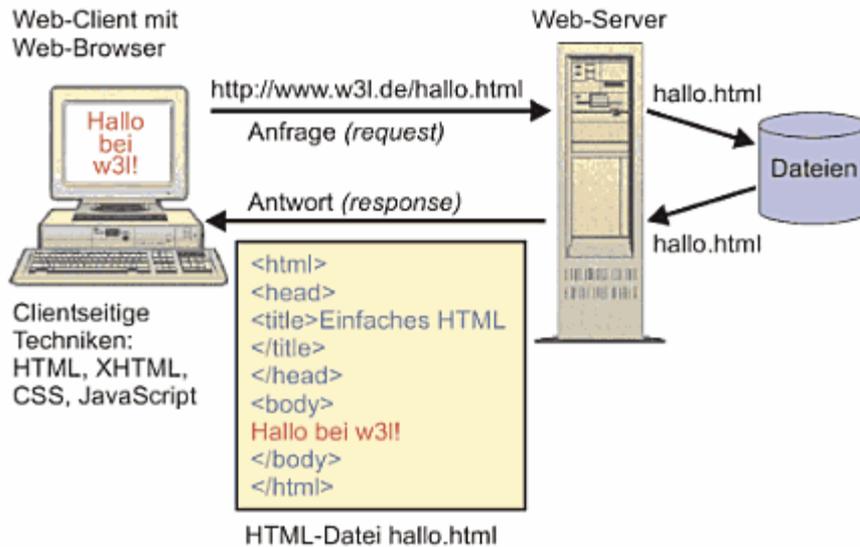


Abb. 2.0-1: Das Bild zeigt, wie die Anfrage eines Web-Browsers nach einer HTML-Datei abläuft. Der Web-Browser fordert die Datei hallo.html an. Der Web-Server liest die Datei von seiner Festplatte und schickt den Dateiinhalt zum Web-Browser, der ihn dann interpretiert und anzeigt.

Dynamisch

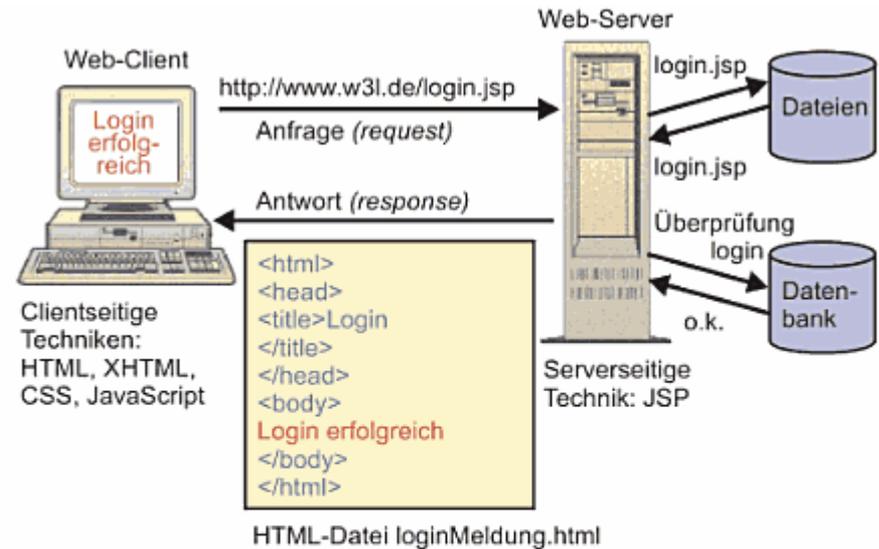


Abb. 2.0-2: Dieses Bild zeigt, wie ein Programm auf dem Server abgearbeitet wird. Der Web-Browser fordert die Datei login.jsp an. Der Web-Server lädt die Datei von seiner Festplatte in seinen Arbeitsspeicher und führt das Programm aus, d.h. überprüft die Login-Daten (Zugriff auf die gespeicherten Daten in einer Datenbank). Das Ergebnis der Überprüfung ist ein HTML-Dokument, das an den Web-Browser gesandt wird.

Benötigte Software

- Java Development Kit (JDK)
- Java Entwicklungsumgebung (optional)
z. B. Netbeans
- Container z. B. Tomcat 
- HTML Editor z. B. Dreamweaver (optional)
 - JSPs erstellen und testen

JSP Elementtypen (Syntax)

- HTML Grundgerüst mit Java-Code Fragmenten
- Java Server Pages tragen die Endung *.jsp
- Zu erkennen durch Markierungen,
sog. Tags `<% ... %>`

Direktive

- Gibt besondere Anweisungen zur Seitenübersetzungszeit an den Container weiter
- Schreibweise `<%@ ... %>`

Beispiel: `<%@ page import =“java.util.*“ %>`

Scriptlet

- Java Code in `<% ... %>`

Beispiel: `<% Float eins = new Float (42.5); %>`

Ausdruck

- Zum Ausgeben von print()-Methoden
- Vorteil:
ist kürzer, Ausgabe nicht explizit angeben
- Schreibweise `<%= ... %>`
- Achtung: Niemals mit Semikolon abschließen!

Beispiel: `<%= Zaehler.getZaehler() %>`

Statt: `<% out.println (Zaehler.getZaehler()); %>`

Deklaration

- Deklaration um Instanzvariablen zu deklarieren
- Scriptlet- und Ausdruckscode sind immer lokale Variablen
- Schließt Variablen wie Methoden mit ein
 - Statische Variablen & Methoden
 - Instanzvariablen & Methoden
- Stehen außerhalb der Service-Methode
- Schreibweise `<%! ... %>`

Kommentare

- Schreibweise `<%-- Kommentar --%>`
- `//` Kommentar funktioniert nicht, wird sonst als Teil der Antwort dem Client angezeigt
- Ausnahme: In Scriptlet oder Deklaration-Tag

Aktion

- Mittel zum erstellen von Objekten und zur Manipulation von JSP
- Bietet die Möglichkeit vorgegebenen Ablauf zu verändern
- Man unterscheidet Standardaktionen und selbst definierte Tags

Beispiel: `<jsp: include page =“foo.html“/>`

Container - Tomcat

- JSP-Servlets besitzen keine main()-Methode
- Container übernimmt diese Aufgabe
- Container bietet:
 - Kommunikationsdienste
 - Stellt die Plattform für JSP dar
 - Deklarative Sicherheit (Deployment Descriptor)
 - Übersetzt Java Code in Servlets (JSP-Compiler)
 - Erzeugt und Löscht Threads
 - Ordnet die URL dem Servlet zu
 - Prüft URL Anfragen auf Rechte
 - Leitet den Lebenszyklus des Servlet

Deployment Descriptor

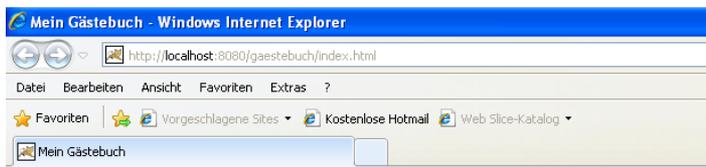
- Eine XML-Datei mit Namen web.xml im Verzeichnis WEB-INF
- Dient zur Konfiguration der Webanwendung
- Tomcat hat ein default DD, liegt unter conf/web.xml

MVC (Model - View - Controller)

- Entwurfsmuster; ideal für Webanwendungen
- Model:
 - Datenmodell; wo die Daten gespeichert werden
- View:
 - Darstellung der Daten auf der Benutzeroberfläche
- Controller:
 - Steuerung, Verarbeitung der vom Benutzer eingegebenen Daten

Gästebuch Beispiel

Die JSP-Datei
deklariert die Klasse
„ObjektDatei.class“



Gästebuch

Name*

E-Mail

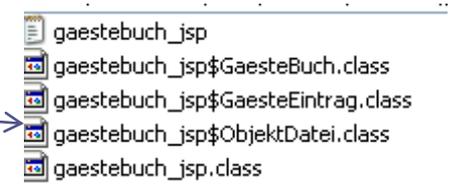
http://
Homepage

Eintrag

[Zur Übersicht der Gästebucheinträge](#)

```
gaestebuch.jsp - Editor
Datei Bearbeiten Format Ansicht ?
color: #333333;
}
</style>
<meta http-equiv="Content-Type" content="text/html" />
<%-- Importieren von Java-Klassenbibliotheken --%>
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
<%@page import="java.text.*"%>

<%-- Deklaration von Java-Klassen --%>
<%!
//Java-Klasse zur Serialisierung und Deserialisierung
class ObjektDatei
{
    private String dateiname;
    //konstruktor
    public ObjektDatei(String dateiname)
    {
        this.dateiname = dateiname;
    }
}
%>
```



Mein Gästebuch

Am 19.02.2010 09:06:19 schrieb:
Christoph Süsens
mit der E-Mail: c.suesens@gmx.de
und der Homepage: http://test
diesen Eintrag: Hallo dies ist ein Test

[Zur Startseite](#)

```
Objekt)
datei =
FileOutputStream(dateiname));
inobjekt);
dateispeichern: "+e);}
}
```

Live Vorstellung des Beispiels

- JDK installieren
- Tomcat installieren
- WebApps installieren

Aufrufen des Beispiels „Gästebuch“

Quellen

- http://en.wikipedia.org/wiki/JSP_container
- <http://books.google.de/books?id=2mnMsnEQwRwC&printsec=frontcover&dq=jsp+f%C3%BCr+einsteiger&cd=1#v=onepage&q=&f=false>
- http://www.ssw.uni-linz.ac.at/Teaching/Lectures/Sem/2000/Leitenmueller/#_Toc501441841
- http://homepages.fh-giessen.de/~hg10013/Lehre/MMS/SS01_WS0102/JSP/aktionen.html#include
- <http://books.google.de/books?id=Yo6BeomN2tgC&printsec=frontcover&dq=servlets+and+jsp+von+kopf+bis+fu%C3%9F&cd=2#v=onepage&q=servlets%20and%20jsp%20von%20kopf%20bis%20fu%C3%9F&f=false>
- <https://wiki.imise.uni-leipzig.de/Themen/WebApp/WebXml>