Aufbau einer Web-Anwendung (JSP o. ASP o. PHP)

von Gael Tchoukio
ntchoukio@yahoo.fr
05.07.2004

1. Einführung in Web-Anwendungen

Eine der am häufigsten gestellten Fragen über Web-Anwendungen lautet: „Ich möchte eine Web-Anwendung aufbauen, weiß aber nicht, wie ich das tun soll."

Für jeden neuen Entwickler von Web-Anwendungen sollte als Grundlage gelten, den Unterschied zwischen Internet und dem World Wide Web zu verstehen.

1.1. Internet

Als Internet wird die Menge aller Rechner (*hosts*) bezeichnet, die über geeigneteVerbindungseinrichtungen unter Nutzung des TCP/IP-Protokolls (*Transmission Control Protocol/Internet Protocol*) miteinander kommunizieren.

1.2. World Wide Web

Besitzt ein Benutzer einen Zugang zum Internet, kann er Informationen an andere Rechner bzw. Benutzer des Internet schicken oder selbst Dokumente aus dem Internet abfragen bzw. erhalten. Der jüngste Dienst im Internet ist das World Wide Web (sehr oft einfach Web genannt). Es ist ein multimediales hypertextbasierendes Informationssystem im Internet.

Die Kommunikation erfolgt zwischen einem Server und einem Client und das Protokoll dafür heißt HTTP (HyperText Transfer Protokoll). Über dieses Protokoll fordert der Client bei einem Server eine ganz bestimmte Datei mit einem Hypertext-Dokument an, die dann vom Server an Client Übertragen wird. Server sind Programme, die permanent darauf warten, dass eine Anfrage eintrifft, die ihren Dienst betreffen. So wartet etwa ein Web-Server darauf, dass Anfragen eintreffen, die Web-Seiten auf dem Server-Rechner abrufen wollen.
Clients sind dagegen Software-Programme, die typischerweise Daten von Servern anfordern. Ihr Web-Browser ist beispielsweise ein Client.

2. Was ist eine Web-Anwendung?

In seiner ursprünglichen Form wurden das World Wide Web (WWW) lediglich als statisches System genutzt, die Inhalte waren also zumeist Dokumente, die eine dynamische Interaktion seitens des Benutzers nicht vorsahen. Seit geraumer Zeit jedoch ist eine Entwicklung offensichtlich, das Internet als Plattform zur Applikationsausführung zu nutzen. Beispiele hierfür sind Online-Benutzerschnittstellen für eine Vielzahl von servergestützten Anwendungen wie E-Mail, Homebanking, die Realisierung von Onlineshops, Internet-Auktionen und anderer Dienstleistungen bis hin zu Officeanwendungen wie Web-Organizern und -Terminkalendern. Die Entwicklung zielt vor allem auf eine Verlagerung der Ausführung von Applikationen auf das Internet hin, die bisher auf Clientrechnern ausgeführt wurden und die die klassischen Möglichkeiten eines statischen WWW-Modells sprengen. Zudem wird

durch den Einsatz leistungsfähiger Webapplikationen auf bestehende IT-Lösungen und -Infrastrukturen ein plattformübergreifender Zugriff ermöglicht.

Web-Anwendungen sind Software-Programme, die geschrieben werden, damit Web-Server auf *User Request* dynamisch reagieren können. Es ist die dynamische Erweiterung von Web-Servern.

Web-Anwendungen bieten viele Vorteilen:

Clientseitig

- Installierung von Software nicht notwendig, ein einfacher Webbrowser reicht, PC, Laptop, PDA, Smartphone, Handy können eingesetzt werden.
- Geringere Kosten
- Zugriff auf eigene Daten überall möglich
- Multiuser- und Netzwerkfähige Anwendungen (Internet und Intranet) viele User können gleichzeitig arbeiten.

Serverseitig

- Anwendungen auf zentralen Applikation Server leicht wartbar
- Realisierung von Verteilung, Transaktionsmanagement
- Sicherheit, Skalierung, Load Balancing und Fail Over möglich

3. Wie sind Web-Anwendungen aufgebaut?

Statische und dynamische Inhalte

Betrachten wir zunächst die prinzipielle Funktionsweise des Internets: Das Web funktioniert gemäß dem Client-Server-Prinzip, wonach der Client dem Server eine Anfrage überstellt, welche dieser bearbeitet und dem Client eine Antwort übermittelt. Beim Anfordern einer HTML-Seite lädt der Server diese von seiner Festplatte und sendet sie an den Client, dieser zeigt die Seite schließlich im Browserfenster an. Solche Webdokumente bezeichnet man auch als *statisch*.

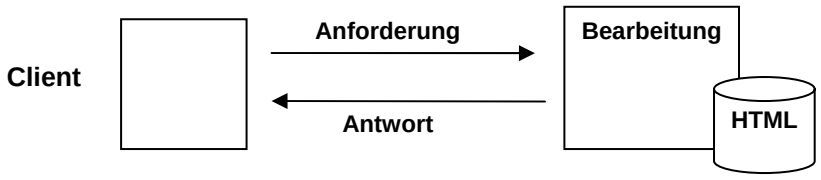


Abbildung 1: Kommunikationsschema für statische Webseiten

Wünschenswert für eine Vielzahl von Anwendungen ist jedoch ein Mechanismus, der es erlaubt, den Inhalt der Webdokumente in Abhängigkeit von Benutzeranfragen zu modifizieren. Solche Webdokumente werden als *dynamisch* bezeichnet.

Client- und serverseitige Dynamik

Hierfür existieren zwei Ansätze. Prinzipiell verfügen moderne Webbrowser über die Möglichkeit, selbst Benutzeranfragen zu qualifizieren. So gibt es eine ganze Reihe von Aufgaben, die unmittelbar beim Client erledigt werden können, beispielsweise Bereichs- und Gültigkeitsprüfungen von Zahlen und Texten in Formularfeldern. Diese Form der Dynamik wird *clientseitig* bezeichnet. Für weiterreichende Webapplikationen reichen die verhältnismäßig einfachen clientseitigen Mechanismen jedoch nicht aus. So muss der Server zum Beispiel für Suchprozesse oder Bestellvorgänge im Internet in der Lage sein, die Anfragen des Clients qualifiziert zu beantworten. Dies wird durch den Einsatz *serverseitiger* Dynamik ermöglicht. Anfragen, die der Client dem Server stellt, werden von serverseitigen Programmen für den Client bedarfsgerecht aufbereitet.

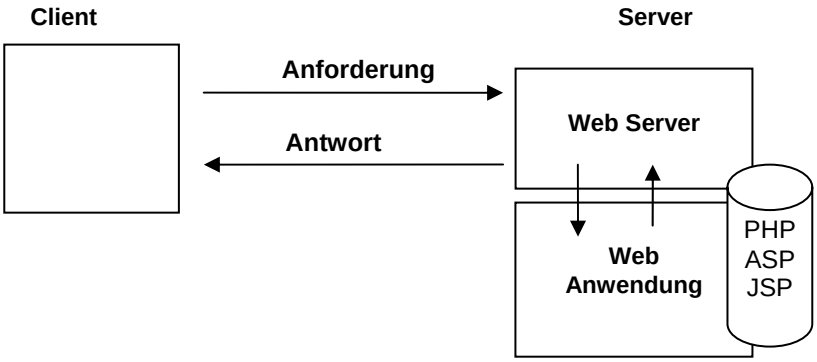


Abbildung 2: Kommunikationsschema für dynamische Webseiten

Web-Anwendungen werden typischerweise in 3 Schichten realisiert:

- Präsentationsschicht (Webbrowser)
- Applikationsserver
- Datenbanksicht

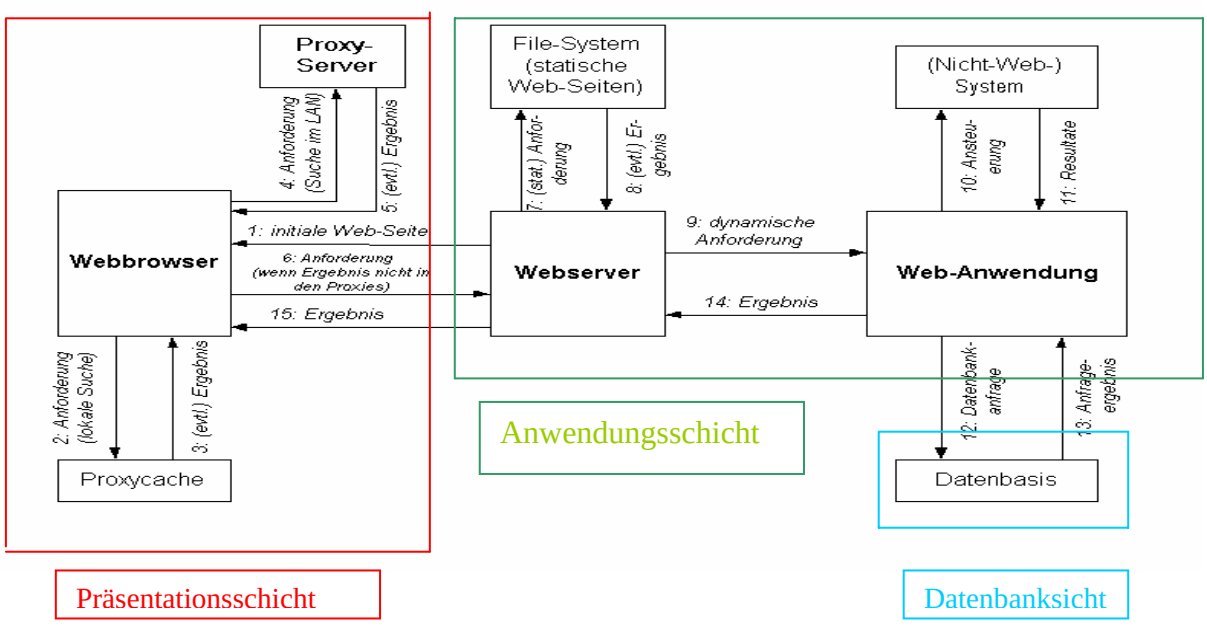


Abbildung 3: Funktionsweise von Web-Anwendungen

Ein *Application Server* (Applikationsserver) ist ein Server für Anwendungsprogramme.
Dies kann ein Server im LAN sein, der von vielen Clients benutzte Anwendungen ausführen.

Oder er führt in einer *Three-Tier-Achitecture* (Drei-Schichten-Modell) die Anwendungslogik/Geschaeftslogik eines Programms aus, vermittelt zu Datenbankservern und erlaubt den Anschluss vieler Clients.

Oder es ist ein Web Application Server (WAS) für Webanwendungen, der für eine Internet- oder Intranet-Anbindung dynamisch HTML-Seiten erzeugt.

4. Eine Web-Anwendung am Beispiel von JSP

4.1. Was sind Java Server Pages?

Java Server Pages, kurz JSP, sind eine verbreitete Möglichkeit Webseiten dynamisch zu gestalten. Insbesondere erreichen JSP eine gute Trennung zwischen Gestaltung der Seiten und Einbindung der Funktionalität. Zusammen mit der erleichterten Einbindung von JavaBeans, lassen sich mit erheblich verringertem Aufwand selbst komplexe Webapplikationen erstellen.

Java Server Pages sind, wie durch die Bezeichnung bereits impliziert, eine Server-seitige Technologie, indem spezielle Tags in gewöhnliche Markup Sprachen, wie HTML, XHTML, XML und ähnliche, eingebunden werden. Üblicherweise werden Dateien, die solche JSP Tags enthalten, mit der Erweiterung .jsp gekennzeichnet, um eine Zuordnung zu einem geeigneten Interpreter zu gewährleisten.

Wird ein solches Dokument angefordert, werden die JSP-Tags, also in spezielle Markierungen gefasste Anweisungen, durch die hieraus generierten Inhalte ersetzt und an den Klienten gesendet.

In der Hierarchie der Java Sprachumgebung sind Java Server Pages eine Erweiterung der Java-Servlet-Technologie.

Java Server Pages bieten aber wesentlich mehr als bloßes Ersetzen einfacher Anweisungen durch generierte Werte. Der Zugriff auf weitere Ressourcen des Servers oder die Weitergabe des Kontrollflusses sind ebenso möglich, wie der Informationsaustausch mit anderen Java Komponenten, beispielsweise Java Beans. Dies garantiert ein hohes Maß an Flexibilität

4.2. Funktionsweise von JSP in Web-Anwendungen

Es bleibt nun zu klären, wie Anfragen an solche Dateien bearbeitet werden. Naheliegend wäre einen Interpreter zu starten, der die JSP Tags auswertet und das so generierte Dokument zurückgibt.

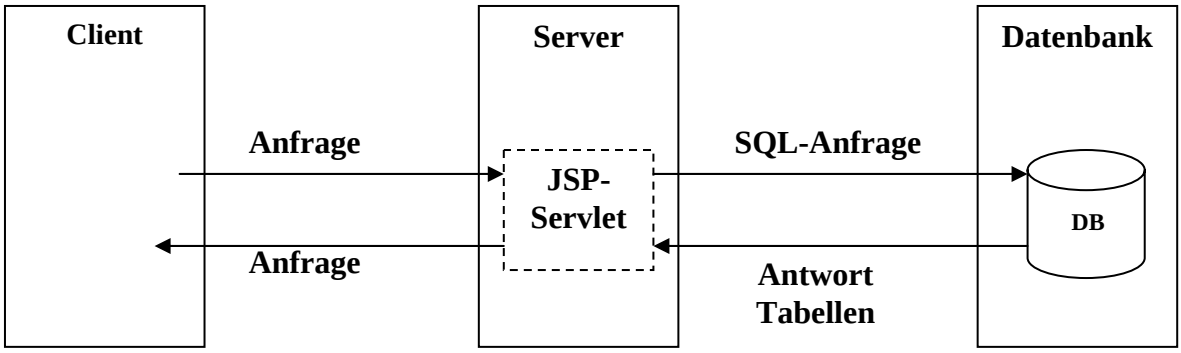


Abbildung 4:

Wann immer eine JSP-Datei erstellt oder verändert wird, sind einige Schritte durchzuführen: Die Datei wird geparst und in Quellcode eines Java-Servlets transformiert. Bei diesem Vorgang wird sämtlicher Text außerhalb der JSP Tags unverändert auf den Rückgabestrom gelegt, die Tags selbst werden durch den entsprechenden Java-Code ersetzt. Falls Anweisungen eine Rückgabe erzeugen, wird auch diese in die Antwort integriert. Der

Quellcode wird schließlich kompiliert und von einem Servlet Container geladen. So ist es letztlich ein Servlet, welches die Anfragen an ein JSP-Dokument beantwortet.

Dieser Transformationsprozess wird ausgelöst durch eine Anforderung, wenn das entsprechende Servlet entweder noch nicht existiert oder aber ein JSP-Dokument jüngeren Datums vorliegt. In jedem anderen Fall wird die Antwort direkt durch das Servlet generiert. Servlets arbeiten recht schnell, da sie sich nach dem Laden im Hauptspeicher befinden und oft ohne Festplattenzugriffe auf Anfragen reagieren können. Problematisch ist nur die erste Anforderung nach Erstellung oder Veränderung der zugehörigen JSP-Datei, da erst die zuvor beschriebene Transformation durchgeführt werden muss. Im Normalfall können aber Werkzeuge des Servlet Containers genutzt werden, um diesen Vorgang zu starten.

4.3. Vorteile von JSP

Wichtig für eine effiziente Entwicklung von Webapplikationen ist vor allem eine möglichst saubere Trennung zwischen der Präsentation und den Inhalten. Ein weiterer Vorteil eines modularen Aufbaus, ist die Möglichkeit Komponenten mehrfach zu benutzen und auch in anderen Projekten nutzbringend einzusetzen oder an veränderte Bedingungen mit minimalem Aufwand anzupassen. Diese drei Hauptmerkmale werden von JSP gut unterstützt, wie im Folgenden dargestellt wird.

Arbeitsteilung

Während bei der Präsentation Designfähigkeiten verlangt werden, kommt es bei der Einbindung von Inhalten oft auf gute Programmierkenntnisse an. Da dies sehr unterschiedliche Fähigkeiten erfordert, lässt sich die Entwicklung von Webprojekten beschleunigen, indem diese Bereiche von unterschiedlichen Gruppen parallel abgearbeitet werden.

Das Programmierteam kann sich gleichzeitig um die Implementierung der Schnittstelle bemühen, ohne sich mit der späteren Darstellung zu befassen.

Wiederverwendung

Eine weitere Folge dieser Trennung ist die Wiederverwendbarkeit der einzelnen Komponenten. Diese werden vom Programmierteam entwickelt, ohne vorherige Kenntnis der Art und Weise wie sie innerhalb des Projekts eingesetzt werden sollen. Bei einer guten Spezifikation, ist lediglich ein gewisses Anforderungsprofil über die Art der benötigten Daten wichtig. Dies schließt die Möglichkeit aus, irgendwelche präsentationsspezifische Anweisungen in den Code einzubetten. Folglich können die Komponenten in der Regel auch in anderen Projekten Anwendung finden.
Ein weiterer sehr wichtiger Aspekt ist, dass es die Java Architektur erlaubt, die Komponenten unabhängig von der eingesetzten Plattform, des Betriebssystems oder des Webservers zu verwenden. Auf diese Weise wird ein Maximum an Flexibilität ermöglicht. Beispielsweise werden in einem Projekt Komponenten eingesetzt, die Informationen aus XML Dateien oder Datenbanken auslesen. Bei einer geeigneten Definition der Schnittstelle sind diese praktisch universell einsetzbar.

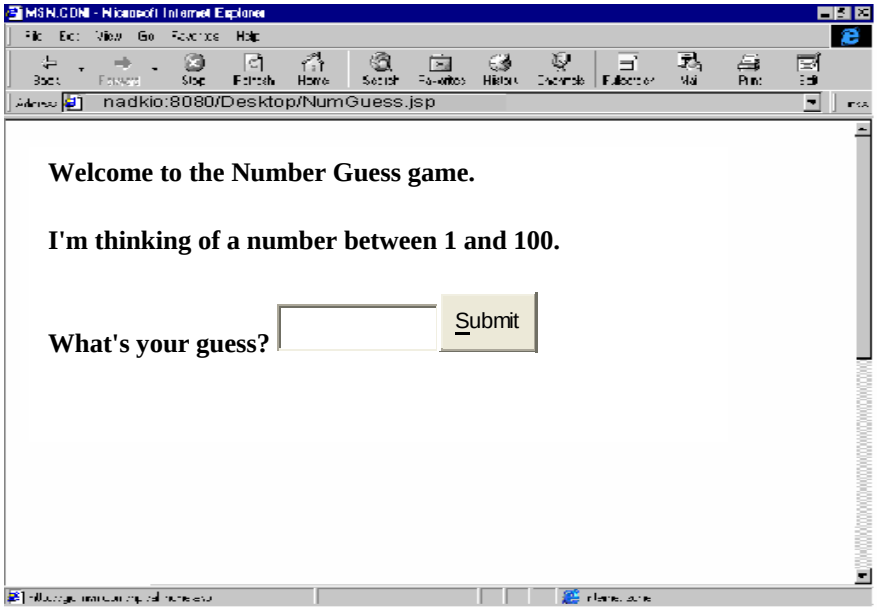
Wartungsfreundlichkeit

Auch der dritte Punkt, die Wartungsfreundlichkeit, wird durch das Trennungskonzept erfüllt. Fehler können rascher lokalisiert werden, denn dass die Programmlogik und der gestalterische Anteil in verschiedenen Quellen liegen, bedeutet einen erheblichen Gewinn an

Übersicht. Kein Designer ist gezwungen Programmcode zu modifizieren, um Veränderungen an der Darstellung vorzunehmen. Kein Programmierer muss an verschiedenen Stellen in HTML Seiten nach Code suchen, um ein Projekt an ein anderes Datenbanksystem anzupassen. Auch an dieser Stelle zeigt sich eine bedeutende Steigerung der Effizienz.

4.4. Beispiel einer Web-Anwendung mit JSP

Kleines Beispiel: Guessing Numbers



5. Literaturverzeichnis

[1] Ben Forta, et al., Java Server Pages Application Development (2001 by SAMS),

[2] Duane K. Fields, Mark A. Kolb, Java Server Pages, Addison-Wesley, 2001

[3] http://www.wdvl.com/Authoring/Tools/Tutorial/
Selena Sol, Introduction to the Web application development, May 31, 1999

[4] http://www-i3.informatik.rwth-aachen.de/teaching/02/proseminar/
B. Frings, M. Jonas, Java Server Pages, Sommersemester 2002