

Dynamische Webseiten

Seminar Medientechnik
30.06.2003

Christoph Metz

Inhalt

- Allgemeine Funktionsweise eines Webserver
- Grundgedanke von dynamischen Webseiten
- Einschub: Dynamische Seitenerzeugung auf der Client-Seite
- Verschiedene Ansätze / Techniken für dynamische Webseiten
- Webseiten mit Datenbankanbindung
- Content Management, Redaktionssysteme

Allgemeine Funktionsweise eines Webservers

- Der Client (i.d.R. ein Browser) schickt übers Internet eine **Anfrage** an den **Webserver**:

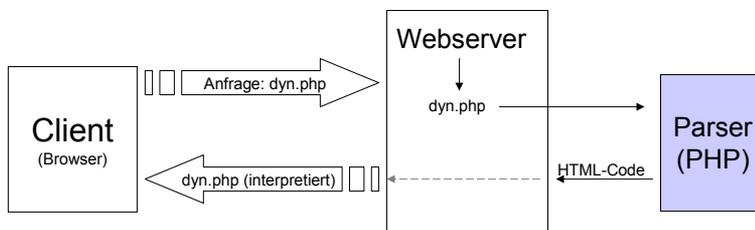
`http://www.server.com/ordner/dateiname.htm`

- Der Server (server.com) schickt das geforderte Dokument (/ordner/dateiname.htm) an die IP-Adresse des Client
- Die Verbindung zwischen Server und Client wird geschlossen
- Das Dokument ist beim Client im Speicher und kann von dort gelesen werden

➤ Probleme:

- Statisches System -> der Benutzer kann nicht einbezogen werden, jeder bekommt die immer gleiche Seite zu Gesicht.
- Keine echte Interaktion möglich
- Aktualisierung der Seite sehr umständlich

Grundgedanke von dynamischen Webseiten



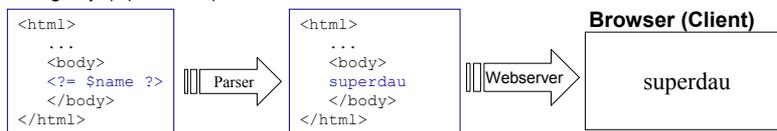
- Die Auswertung und Erstellung der Seite findet komplett auf dem Server statt
- Der Client bekommt weiterhin nur statischen HTML-Code zu Gesicht

- Prinzip: Dokumente werden vom Webserver erst zum Zeitpunkt der Anfrage aus zusätzlichen Anfragedaten (vom Client), oder anderen externen Informationen (z.B. Datenbank), dynamisch erstellt.

Grundgedanke von dynamischen Webseiten

- Der Client kann mit der Anfrage zusätzliche Daten schicken:
 - Cookies (beim Client unsichtbar gespeichert, automatisch vom Browser verschickt)
 - Übergabe von Variablen in der URL („GET“)
`http://www.server.com/ordner/dyn.php?name=superdau`
 - Übergabe von Daten durch ein „POST“-Formular
- Der Server kann diese Daten abfragen und daraus dynamisch eine Seite bauen.
- Beispiel (PHP-Syntax):

Anfrage: `dyn.php?name=superdau`



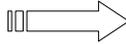
Grundgedanke von dynamischen Webseiten

- Auf dem Server liegt keine statische Datei mehr, sondern im Idealfall eine Seitenbeschreibung ohne Inhalt.
- Abhängig von verschiedenen möglichen Datenquellen (je nach Anwendung) wird die Seite erzeugt und an den Client als statisches Dokument geschickt
- Mögliche externe Datenquellen:
 - Datenübergabe vom Client durch ein Formular
 - Beim Client gespeicherte Cookies
 - Diverse Daten aus einer HTTP-Anfrage, z.B. IP-Adresse des Client (ermöglicht z.T. Lokalisierung des Users -> regionale Seite)
 - Daten aus einer Datenbank
 - Umgebungsdaten wie Uhrzeit, Serverauslastung, etc..
 - Alle weiteren vom Webserver aus zugänglichen Daten (beliebig viele)
- Vorteil: Programmcode ist nicht öffentlich zugänglich - der User bekommt nur die Ausgabe des Programms zu Gesicht

Einschub: Dynamische Seitenerzeugung auf der Client-Seite

- Es gibt auch Ansätze, eine Seite auf der Clientseite dynamisch zu erzeugen: **DHTML** (= Dynamic HTML -> Erweiterung von HTML durch Javascript und CSS)

```
<script>
  var cTime = new Date();
  var cYear = cTime.getFullYear();
  document.write(cTime);
</script>
```



2003

- Mit Javascript lassen sich z.B. Cookies schreiben und lesen - auf diese Weise lässt sich z.B. ein kompletter Online-Shop realisieren.
- Dynamische Art der Darstellung durch externe CSS-Definitionen
 - Achtung: Dieser Ansatz fällt nicht unter den Begriff „Dynamische Webseiten“ - strenggenommen sind alle DHTML-Seiten nur statische Dokumente
 - Der (immer gleiche) Programmcode wird komplett zum Client übertragen und dort ausgeführt

Einschub: Dynamische Seitenerzeugung auf der Client-Seite

- Vorteile:
 - Es muss keine Verbindung zum Server aufgebaut werden
 - Berechnungen können auf dem Client ausgeführt werden
 - Eingabefehler können abgefangen werden, bevor Daten zum Server übermittelt werden
 - Einsparung von unnötiger Client-Server Kommunikation
 - Geringere Server-Auslastung (Traffic, sowie Rechenzeit)
 - Schnellere Ausführung
- Unmittelbare Veränderungen auf der Webseite (nach einer User-Interaktion) sind nur durch dynamischen Code auf der Client-Seite möglich - abgesehen von Applets oder Plugins, sind solche Interaktionen nur mit DHTML möglich
 - In der Praxis kommt oft eine Kombination zum Einsatz - die Seite wird sowohl beim Server als auch beim Client dynamisch erzeugt, d.h. der vom Server dynamisch erzeugte Code enthält noch Teile, die erst beim Client ausgewertet werden (können,sollen,dürfen)...
 - Nachteil: Der Quellcode wird offen, also von jedem les- und kopierbar zum Client geschickt

Verschiedene Ansätze / Techniken für dynamische Webseiten

- Unterscheidung zwischen CGI und sog. „Serverpages“
- CGI => Common Gateway Interface
 - Eine Schnittstelle des Webservers zum Aufruf beliebiger externer Programme
 - Das Programm muss als Rückgabe, eine komplette gültige HTML-Seite liefern
- Serverpages
 - Programmcode wird direkt in die HTML-Seite eingebunden

CGI (Perl)

```
#!/usr/bin/perl

%in = %myCGI::getData ();

print "Content-type: text/html\n\n";

print "<html>";
...
print "<body>";
  print $in{'name'};
print "</body>";
print "</html>";
```

Serverpages (PHP)

```
<html>
...
<body>
  <?= $name ?>
</body>
</html>
```

Ansätze für dyn. Webseiten - CGI

- Über die CGI-Schnittstelle kann ein beliebiges Programm auf dem Server aufgerufen werden, dass allerdings eine gültige HTTP-Antwort (meistens ein HTML-Dokument mit(!) HTTP-Header) zurückliefern muss
- Sehr verbreitet ist die Skript-Sprache PERL, daher spricht man oft von CGI-Skripts.
- Bei größeren Webseiten mit viel Rechenaufwand (Routenplaner, Suchmaschine) kommen allerdings nur vorkompilierte Programme zum Einsatz
- Bei vielen Skriptsprachen wie PERL oder Python existiert auch ein Ansatz diese Sprachen als Serverpages zu verwenden (Embedded Perl, Python Server Pages), oder aber spezielle Web-Funktionen, die die Handhabung wesentlich vereinfachen (z.B. bei PERL das Modul CGI.pm)

Ansätze für dyn. Webseiten - Serverpages

- SSI - Server Side Includes
- PHP - Hypertext Preprocessor
- ASP - Active Server Pages (Microsoft)
- JSP - JavaServer Pages

Ansätze für dyn. Webseiten - SSI

- Einfachste Stufe von dynamischen Webseiten
- Sehr geringer Funktionsumfang <-> sehr schnell

```
<html>
...
<body>

Du bist der <!--#exec cgi="/cgi-bin/counter.cgi" -->. Besucher
...
Letzte Änderung: <!--#if "10" != "20" echo var="LAST_MODIFIED" -->
<!--#include file="footer.html" -->

</body>
</html>
```



```
Du bist der 10. Besucher
Letzte Änderung: Fri Jul 21 21:24:48 1995
© 2003 by Mr. Nobody
```

- Kann von Sprachen wie PHP, ASP oder JSP vollständig ersetzt werden

Ansätze für dyn. Webseiten - PHP

- Beginn 1995 als 'Personal Home Page Tools' - eine Zusammenstellung von Perl-Skripts
- Entwicklung zu einer in C geschriebenen Hypertext-Preprocessing-Engine. Mitte 2000 wurde Version 4.0 veröffentlicht
- PHP ermöglicht die Entwicklung komplexer Anwendungen, ist leicht erweiterbar und bietet Unterstützung für viele Datenbanken, Protokolle und APIs
- Speziell für den Web-Einsatz zugeschnitten, alle nicht programmspezifischen Dinge übernimmt PHP automatisch
- Leicht erlernbar
- Ab Version 4.0 sehr schnelle Verarbeitung

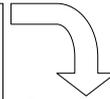
Ansätze für dyn. Webseiten - PHP

- Viele Standardaufgaben werden von PHP automatisch übernommen (-> der Programmierer kann sich auf sein Programm konzentrieren).
- Beispiele:
 - Setzen und Auslesen von Cookies: `setcookie("mycookie", $value)`
 - Viele Funktionen, für Standardaufgaben in der Web-Entwicklung, z.B.
`$string = strip_tags($string, '<a><i><u>');`
=> entfernt alle HTML-Tags aus einem String, außer die angegeben
 - Automatisches Auslesen der vom User übermittelten Daten
 - Speicherung von übermittelten Dateien
 - Integration von verschiedenen Datenbanken
 - Unterstützung zum Schreiben von vielen Formaten:
 - JPEG
 - GIF
 - PDF
 - Flash
 - (endlose) Erweiterungen, z.B. eingebaute Funktionen für verschiedenste Protokolle (FTP, POP3, IMAP)

Webseiten mit Datenbankanbindung

- Bei vielen Web-Anwendungen ist eine Datenbank unerlässlich
- Am weitesten verbreitet bei kleinen bis mittelgroßen Projekten ist die freie Datenbank MySQL
- Beispiel einer Datenbankanfrage in PHP:

```
<? db_connect(); ?>
<html>
...
<body>
<table><tr><td>Name</td><td>Vorname</td><td>GT</td></tr>
<?
$result = mysql_query("SELECT * FROM users");
while ($db_user = mysql_fetch_array($result)) {
    print "<tr>";
    print "<td>$db_user[name]</td>";
    print "<td>$db_user[vorname]</td>";
    print "<td>$db_user[geburtstag]</td>";
    print "</tr>";
}
?>
</table>
</body>
</html>
```



Browserdarstellung

Name	Vorname	GT
Franz	Schmidt	20.03.1932
Otto	Meyer	30.03.1959
Muster	Mann	01.01.2001
	...	

Content Management, Redaktionssysteme

- Die Verwendung von Datenbanken bei Webseiten ermöglicht den Einsatz sog. Content-Management-Systeme (CMS)
- Der Inhalt einer Seite ist dabei nicht mehr im HTML-Dokument festgelegt, sondern wird in einer Datenbank abgelegt - die HTML-Datei definiert nur noch die Art der Darstellung.
- Auf diese Weise lassen sich Änderungen an einer Webseite schnell vollziehen, ohne dass man irgendwelche Dateien ändern müsste.
- Ein CMS braucht dann aber noch ein sog. Backend, das für den Webmaster, bzw. die Redakteure zugänglich ist, um Änderungen an der Datenbank vorzunehmen

