

Doing Web Apps

Installation eines WAMP-Systems

Autor:	Rüdiger Marwein
Letzte Änderung:	2012-10-18
Version:	0.9
Copyright:	2012. Alle Rechte vorbehalten

Dieses Dokument darf - mit Nennung des Autoren - frei vervielfältigt, verändert und weitergegeben werden.

Der Inhalt ist sorgfältig recherchiert, mit dem Dokument ist jedoch keinerlei Garantie auf Fehlerfreiheit gewährleistet.

Dieser Inhalt ist unter einem Creative Commons Namensnennung Lizenzvertrag lizenziert. Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by/2.0/de/> oder schicken Sie einen Brief an Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Inhaltsverzeichnis

1.Zutaten.....	3
2.Installation des XAMPP.....	4
2.1.Wichtige Verzeichnisse und Dateien	4
2.2.Erweiterte Konfigurationshinweise.....	5
3.Begriffswelt.....	6
3.1.Port.....	6
3.2.Server.....	6
3.3.Webserver.....	6
3.4.DocumentRoot.....	6
3.5.Domain.....	6
3.6.VirtualHost.....	6
3.7.localhost.....	6
3.8.DirectoryIndex.....	6
3.9.Mime-Types.....	7
3.10.Dateiendungen.....	7
3.11.Schrägstrich - Slash.....	7
4.PHP Konfiguration.....	8
4.1.magic_quotes.....	8
4.2.register_globals.....	8
4.3.variables_order.....	8
4.4.error_reporting.....	9

1.Zutaten

WAMP	Windows - Apache - MySQL - PHP
LAMP	Linux - Apache - MySQL - PHP
MAMP	Mac - Apache - MySQL - PHP

Bezugsquellen:**Apache Webserver**

<http://httpd.apache.org/download.cgi>

MySQL Datenbank-Server

<http://dev.mysql.com/downloads/>

PHP Serverseitige Scriptsprache

<http://www.php.net/downloads.php>

Hier nur das zip-package downloaden

Software-Paket XAMPP

<http://www.apachefriends.org/de/xampp.html>

(Windows, Linux, Mac)

Die Installation des **XAMPP Software-Pakets** ist einfach und zweckmäßig für eine lokale Testumgebung. Für die Installation der Einzelkomponenten stehen anderweitig Dokumente der jeweiligen Hersteller zur Verfügung.

Bspw.: <http://www.php.net/manual/de/install.windows.php>

Auf aktuellen Linux-Systemen wie Ubuntu oder CentOS stehen Pakete zur Verfügung, die Installation und Konfiguration großteils übernehmen. Die Komponenten sind:

- MySQL Server + Client
- Apache Webserver
- PHP CLI
- mod_php für Apache
- phpmyadmin

Im Folgenden wird von einer XAMPP Installation ausgegangen.

2.Installation des XAMPP

Die Installationsdatei der Software ist unter der eingangs angegebenen URL beziehbar.

Hinweis für Windows Vista und 7: Es muss ein Verzeichnis außerhalb von „Programme“ gewählt werden. Ist UAC (Benutzerkontensteuerung) aktiviert, lassen sich die Serverdienste sonst nicht über die Benutzeroberfläche starten und stoppen.

Das Setup kann linear durchlaufen werden. Installieren Sie Apache und MySQL als Dienst.

Nach abgeschlossener Installation befindet sich auf dem Rechner ein ad-hoc lauffähiges Windows-Apache-MySQL-PHP-System. PHP läuft als Modul innerhalb des Apache Webservers.

Nach der Installation ist das System unter `http://localhost/`¹ aufrufbar.

Hinweis für Skype-Nutzer: Skype nutzt selbst den Port 80. Das verhindert den lokalen Betrieb eines Webservers. Zu **deaktivierende** Option unter „Erweitert / Verbindung“:

Ports 80 und 443 als Alternative für eingehende Verbindungen verwenden.

Das Verwaltungsprogramm für die MySQL-Datenbank ist unter `http://localhost/phpMyAdmin` aufrufbar.

Benutzer: root

Passwort leer.

2.1.Wichtige Verzeichnisse und Dateien

Ausgehend von `xampp` Installationsverzeichnis:

<code>apache/conf</code>	Konfigurationsdateien des Apache Webservers
<code>apache/conf/httpd.conf</code>	Hauptkonfigurationsdatei
<code>apache/conf/extra/httpd-vhosts.conf</code>	Domain-Konfigurationsdatei
<code>apache/modules</code>	Ablageort für Apache-Module
<code>htdocs</code>	Ablageort für Webseiten ²
<code>mysql/data</code>	Datenbank-Dateien
<code>php/php.ini</code>	PHP Konfigurationsdatei

¹ Die Web-Adresse `localhost` ist der Webserver auf dem eigenen Rechner mit der IP `127.0.0.1`. Unter diesem Namen ist er nur lokal zu erreichen. **Ist eine Internetverbindung vorhanden, ist er über die vom Provider zugewiesene IP-Adresse im Internet sichtbar.** Ist eine Firewall vorgeschaltet kann der externe Zugriff unterbunden werden.

² „htdocs“ steht für **HyperText Documents**. In diesem Verzeichnis liegen die `.html` bzw. `.php`-Dateien, Bilder und was sonst über den Webserver nach außen gelangen darf.

php/PEAR	Ablageverzeichnis für PEAR-Bibliotheken
php/ext	Ablageort für PHP-Module
phpMyAdmin	Verwaltungsprogramm für die MySQL-Datenbanken mit Web-Oberfläche
C:\Windows\system32\drivers\etc\hosts	Bekannte Host-Namen und zugeordnete IP-Adresse

PHP-Module können in der Datei php.ini hinzugefügt werden, indem das Kommentarzeichen vor der jeweiligen Zeile entfernt wird.

Apache-Module können in der Datei httpd.conf hinzugefügt werden, indem das Kommentarzeichen vor der jeweiligen LoadModule-Zeile entfernt wird.

2.2.Erweiterte Konfigurationshinweise

Soll ein anderes Verzeichnis als xampp/htdocs das Web-Verzeichnis sein, so müssen die Zugriffsrechte für dieses Verzeichnis analog zu htdocs in der httpd.conf gesetzt werden. Dies gilt auch für abweichende VirtualHosts.

Nach einer Änderung an den Apache-Konfigurationsdateien ist ein Neustart des Webserver erforderlich.

Benannte VirtualHosts (wie bspw http://meinserver.local) müssen in der hosts-Datei hinzugefügt werden um im Browser genutzt werden zu können.

Gegebenenfalls muss unter Windows der DNS-Cache geleert werden. Dazu über die cmd Kommandozeile (mit Administratorrechten gestartet) eingeben:

```
ipconfig /flushdns
```

3.Begriffswelt

3.1.Port

Ein Port ist wie ein Namensschild am Briefkasten. Durch die Port-Nummer weiß das Host-System, welcher Server der Empfänger für Daten ist, die über das Netzwerk eintreffen.

Welcher Port für welche Dienste standardisiert wurde, findet man unter <http://www.iana.org/assignments/port-numbers>

Ein Server kann auf jedem beliebigen Port laufen, ungeachtet der Standards, jedoch ist der Zugriff mit „Standardprogrammen“ dann erschwert.

3.2.Server

Ein Server ist ein Programm, das auf einem Computer gestartet wurde und Netzwerk-Verbindungen entgegennehmen und beantworten kann. Ein Server belegt je nach Typ einen Port. Oft werden Server als Systemdienste gestartet, die durchgehend laufen.

3.3.Webserver

Der Apache ist einer von viele Webservern. Ein Webserver verwendet standardmäßig den Port 80 (bzw 443 für SSL/https) und spielt HTML-Seite und Datendateien mittels des HTTP-Protokolls aus.

Weitere Webserver sind bspw LigHTTP oder NGINX.

3.4.DocumentRoot

Das Start-Verzeichnis einer Domain. Beispielsweise C:\xampp\htdocs\. Hierunter liegen die Dateien die unter <http://localhost/> abrufbar sind.

3.5.Domain

Eine Domain ist der Name, der für einen Rechner im Internet (bzw dessen IP-Adresse) registriert wurde. Wird Beispielsweise die URL <http://www.google.com> aufgerufen, wird www.google.com über DNS (Domain Name Service) in die IP 209.85.149.105 aufgelöst.

3.6.VirtualHost

Ein virtueller Host ist eine zusätzliche Adresse, unter der der Webserver erreichbar sein soll. Über virtuelle Hosts können mehrere Domains durch einen Webserver verwaltet werden. Ein VirtualHost kann auf ein anderes Verzeichnis zeigen und eine andere Konfiguration aufweisen.

3.7.localhost

Der lokale Computer. „localhost“ ist der Name der zugehörigen IP-Adresse 127.0.0.1, was

den lokalen Rechner meint.

3.8. DirectoryIndex

Legt fest welche Datei aufgerufen wird, falls in der URL keine Datei angegeben wurde. Oft ist dies index.html, die Konfiguration ist jedoch beliebig. Der xampp-Apache sucht nach Dateien in der folgenden Reihenfolge:

```
index.php index.php4 index.php3 index.cgi index.pl index.html  
index.htm index.shtml index.phtml
```

3.9. Mime-Types

Mime-Types (eigentlich MIME Content-types) sind Dateitypen mit einer Kennung der Form typ/name. Beispielsweise application/x-pdf für PDF-Dokumente. Bekannte Mime-Types findet man hier: <http://www.iana.org/assignments/media-types/>

3.10. Dateiendungen

Dateiendungen stehen am Ende des Dateinamens hinter dem Punkt. Bspw. Index.html. Der Webserver kennt diverse Dateitypen. Der Dateityp application/x-httpd-php mit der Endung „php“ ist an das PHP-Modul geknüpft. Das heißt, alle Dateien die auf .php enden, werden vor der Auslieferung durch das php-Modul verarbeitet bevor die Datei ausgeliefert wird.

Wichtig: Damit Dateiendungen unter Windows sichtbar sind, muss man folgende Einstellung machen bzw entfernen (empfohlen):

Ordneroptionen / Ansicht → Erweiterung bei bekannten Dateitypen ausblenden

3.11. Schrägstrich - Slash

Auf Windows-Systemen wird zur Trennung von Verzeichnissen der Backslash (der umgekehrte Schrägstrich - AltGr+?) verwendet. Vom Webserver wird immer ein normaler Schrägstrich in der URL verwendet. Dies stammt aus Unix-Systemen, da dort für die Ordnerstruktur durchgängig der Slash (Schrägstrich) verwendet wird.

Ist bei der Angabe von Pfaden zu beachten.

4.PHP Konfiguration

Dieses Kapitel beschreibt ein paar wichtige Eigenschaften von PHP, die über die php.ini geregelt sind. Kennt man sie nicht, steht man oft vor seltsam anmutenden Problemen.

4.1.magic_quotes

Magic_quotes fügen Strings automatisch Entwertungszeichen (Backslash -> \) hinzu. So wird aus „Dies ist Peter's "guter" Hund“ folgende Zeichenkette:

„Dies ist Peter\'s \'guter\' Hund“

Das kann praktisch sein, da MySQL-Feldinhalte beim Einfügen ebenfalls Zeichenketten sind und ein " oder ' mittendrin die Kette an der falschen Stelle beenden würde (Sicherheitsrisiko).

Tipp: **Off** setzen und sich gewissenhaft selbst darum kümmern.

4.2.register_globals

Werte, die als Parameter über die URL (GET) oder Werte von Formularen (meist POST) befinden sich in Arrays, bei denen der Name der Schlüssel ist. Z.B.

http://www.google.com/?q=GoogleSuche

implizit: \$_GET['q'] = 'GoogleSuche'.

PHP kann diese Schlüssel automatisch zu Variablen machen, so dass \$q einfach vorhanden ist.

Tipp: **Off** setzen, da es eine der häufigsten Fehlerquellen und **Sicherheitsrisiken** ist.

Ist seit PHP4 initial „Off“. Wird in PHP6 abgeschafft.

4.3.variables_order

Dies spielt bei aktiviertem **register_globals** eine Rolle, weil so festgelegt wird, welche Variablen aus welchem globalen Eingabewerte-Array höhere **Priorität** haben. Es macht aber auch klar, **aus welchen Quellen** man Werte erhalten kann.

Die normale Reihenfolge ist

- **Session** **\$_SESSION** (im Server gespeicherte Variablen des Benutzers) überschreibt
- **Cookie** **\$_COOKIE** (Parameter in Datei auf Benutzerseite) überschreibt
- **POST** **\$_POST** (Formular-Parameter) überschreibt
- **GET** **\$_GET** (URL-Parameter) überschreibt
- **Environment** **\$name** (aktuell deklariert)

Tipp: **EGPCS** wie vordefiniert.

Hinweis: Dass man so mit `register_globals` ganz schnell den Überblick über seine Variablen verliert, ist **offensichtlich**. Es sind immerhin **5 Scopes** (Anwendungsbereiche / Bereiche / Sichten) zu berücksichtigen, von denen 2 (Session und Cookie) auch **über längere Zeit** und **über alle Seiten** hinweg vorhanden sind. Sollten diese Variablen sich in irgendeiner Konstellation gegenseitig überschreiben, kann dies in **mehreren Tagen Fehlersuche** enden.

Weiter können durch `register_globals` Anwender böswillig Variablen in den Code einschleusen. Wurde eine explizite Deklaration vergessen, können Sicherheitslücken entstehen die bis zur „feindlichen Übernahme“ des Betriebssystems reichen.

4.4.error_reporting

Die xampp-Installation beinhaltet eine zu tolerante **Fehlerbehandlung**.

Will man externe Projekte einsetzen ist dies u.U. sinnvoll, weil die Autoren manchmal auf **Notices (geringfügige Fehler)** keine Rücksicht nehmen und die einige Projekte sonst nicht lauffähig wären; oder php4-Kompatibilität benötigt wird.

Wir schauen uns einmal in der `php.ini` um, um diese Einstellung zu verstehen.

Dazu öffnen wir die `php.ini`.

Darin suchen wir „`error_reporting`“ und schauen uns den (nicht-auskommentierten) Eintrag an:

```
error_reporting = E_ALL & ~E_NOTICE & ~E_STRICT;
```

Damit entfallen Fehlermeldungen für uninitialisierte Variablen, nicht belegte Array-Schlüssel und einige andere, die ebenfalls unter dem Schlagwort „**Notiz, aber kein Fehler**“ oder „**Ganz strikt gesehen ist das nicht gut**“ laufen.

Über diesem Eintrag findet man eine Reihe Beispiele, wie man diese Einstellungen noch vornehmen könnte.

Wir ändern das `error_reporting` zu

```
error_reporting = E_ALL;
```

In einem PHP-Script kann man diesen Wert **nachträglich** noch **manipulieren** durch bspw. die Zeile:

```
<?php
    ini_set('error_reporting', E_ALL);
?>
```

Es gibt 3 wichtige Formen von Fehlern in PHP:

Notice: geringfügiger Fehler. Script läuft dennoch (meistens)
Warning: ungünstiger Fehler. Script versucht weiterzulaufen
Fatal: schwerwiegend. Script bricht sofort ab.

Nach jeder Änderung der `php.ini` muss der Apache Webserver neu gestartet werden.