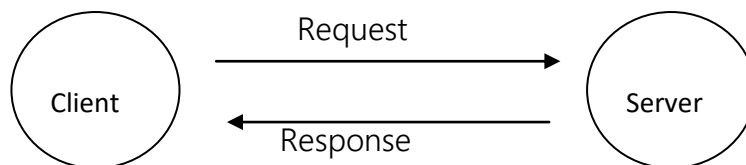




Internet

Client/Server-Struktur



Ein Client ist ein Programm das einen Dienst (Service) von einem anderen Gerät anfordert. Das kann das Schicken von Daten oder auch eine mechanische Funktion sein. Ein Server ist ein Programm, das eine Funktion auf Anforderung ausführt. Die Anforderung des Dienstes heißt Request (Service-Request), das Erbringen des Dienstes wird Response genannt. Der Client ist also ein Requester, der Server ein Responder.

Ihr Nachbar spielt READ-Responder, sie sind der READ-Requester. Der Dienst heißt READ, er beinhaltet das Lesen des ersten Buchstabens auf einer Seite.

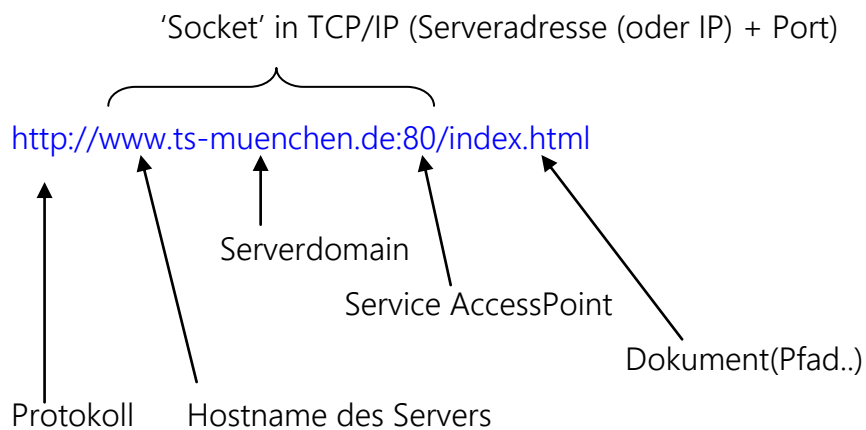
- 1) Sie (Client) sprechen einen READ-Request aus
- 2) Ihr Nachbar (Server) kann den Empfang erst mal quittieren oder nicht (beides gibt's)
- 3) Ihr Nachbar führt den READ-Dienst aus, er liest den Buchstaben
- 4) Ihr Nachbar schickt einen READ-Response. Das bedeutet erst mal nur, daß er den READ gemacht hat. Ob er ihnen den Buchstaben auch sagt, hängt von der Definition des READ-Dienstes ab.

HTTP-Request

Im Internet (und allen anderen TCP/IP-Netzen) wird nun bei Dienstanforderungen angegeben, welches Protokoll zur Kommunikation benutzt wird. Das Protokoll bestimmt dann auch, wie die Dienste im Detail aussehen. Der Standard im Internet ist die Aussprache eines HTTP-Request durch eine Client-Software (ein Browser) und dessen Beantwortung durch den HTTP-Response des Serverprogramms. Der Transport durch TCP/IP geschieht standardmäßig über SAP („Port“) 80.

(SAP unklar ? -> <https://portal.ts-muenchen.de/Portaldateien/Skripten/ports.pdf>)

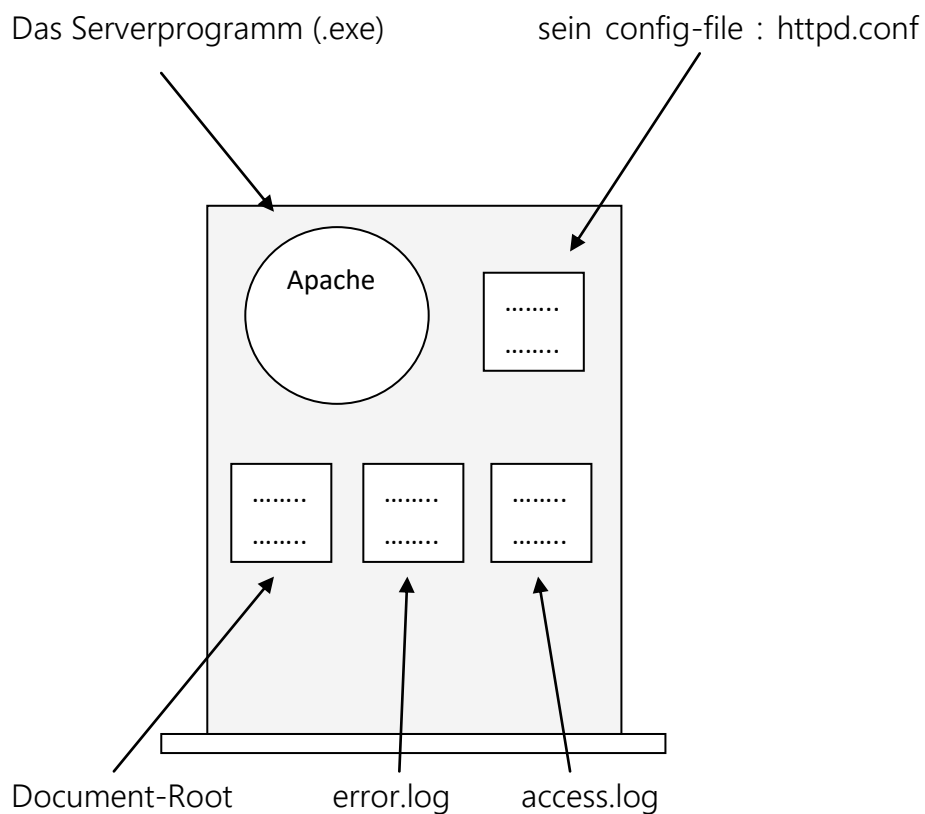
Das Adressformat für einen korrekten HTTP-Request sieht so aus :



Dieses Format heißt Uniform Resource Locator (URL). www ist hier kein Protokoll oder so, der Server hat diesen Hostnamen !

Apache-Server

Als Serverprogramme sind viele Alternativen möglich, der Windows-Server (bei den Serverbetriebssystemen von Microsoft enthalten) heißt IIS (Internet Information Server). Am meisten verbreitet ist jedoch eine OpenSource – Variante, der Apache Server. Die wesentlichsten Komponenten und Dateien sehen Sie hier :



In Document-Root (Verzeichnis) stehen alle Webseiten, in error.log werden alle Zugriffsfehler, in access.log alle Zugriffe (z.B. Seite nicht vorhanden) aufgezeichnet

httpd.conf

Zuerst eine kleine Nebenbemerkung : die meisten Serverprogramme enden im Dateinamen mit einem „d“. Das kommt aus UNIX-Zeiten, da werden Programme, die nicht direkt mit dem Benutzer am Rechner kommunizieren, „Dämonen“ genannt. Apache ist ein http-Dämon. Daher das „d“. (Man spricht das „http-,di'“, also das ‚d‘ quasi separat).

Sein Config-file httpd.conf ist ein normales ASCII-File, das mit dem Windows-Editor (rechte Maus „Bearbeiten“) bearbeitet werden kann. Sie finden es im Verzeichnis von apache (wahrscheinlich xampp/apache/..) unter conf.

Heißer Tipp : Machen Sie immer (!!) vorher eine Sicherungskopie, bevor sie rumbasteln.

Das sind alles Kommentarzeilen

```
#  
# ServerAdmin: Your address, where problems with the server should be  
# e-mailed. This address appears on some server-generated pages, such  
# as error documents. e.g. admin@your-domain.com  
#  
ServerAdmin admin@localhost  
..
```

Hier wird der Parameter „ServerAdmin“ gesetzt

Apache



Sie können das auf ihrem Client/Server-System aus dem Windows-Praktikum machen, oder auf jedem anderen beliebigen Windows. Es gefährdet den Rechner nicht.

Zuerst das Apache-Paket „XAMPP“ installieren.

Holen Sie hierzu das Quellfile WAMPP.zip vom Portal.

Erzeugen Sie auf ihrer Platte ein Verzeichnis (z.B. „webserver“)

Entpacken sie das zip-file in das neue Directory:

Stellen sie für das „Control-Panel“ (xampp-control : im Verzeichnis suchen !) eine Verknüpfung auf Ihrem Desktop her.



Im Konfigurationsfile des Apache-Servers httpd.conf stellen sie den Parameter DocumentRoot auf /xampp/webseiten. (Sicherungskopie von httpd.conf vorher nicht vergessen).

In diesem Verzeichnis sucht apache nacher angeforderte Seiten.

Dieses neue Directory „webseiten“ müssen sie natürlich auf der Platte auch anlegen (in der gleichen Hierarchieebene, in der sich das alte DocumentRoot-Directory : „htdocs“ befindet).

Erzeugen sie ein minimales HTML-File start.html in ihrem neuen DocumentRoot, das die Meldung „hallo, ich bin dein webserver“ ausgibt :

```
<html>
<body>
hallo, ich bin dein webserver
</body>
</html>
```



Nebenbemerkung : Erzeugen einer Datei, die nicht .txt heißt :
Rechte Maus auf ein Directory -> „Neue Textdatei“ . Dann
„Bearbeiten“. Und jetzt wird's schwierig : das file heißt zunächst
name.txt. ch mach jetzt so weiter : Nach dem Bearbeiten unter
einem anderen (!) Namen speichern (z.b. test.html), mit der
richtigen Extension (hier .html). Und nun nicht einfach
„Speichern“, sonst heißt das Ding test.html.txt. Nach
„Datei/Speichern“ den richtigen Dateityp wählen : „Alle Dateien“

Wie immer nach Änderungen der Konfigurationsdatei : Aoache
neu starten (Control Panel)

Dann im Browser aufrufen : <http://localhost/start.html>



Nun noch der SAP (Port).

Suchen Sie im httpd.conf den Parameter „listen“.

Den stellen Sie von 80 auf einen anderen Port, z.b. 11180

Nun wieder Restart von Apache, und jetzt versuchen Sie, ihre Seite wieder im Browser aufzurufen. Da muß jetzt ein korrekter URL angegeben werden, mit dem richtigen Socket ;-)