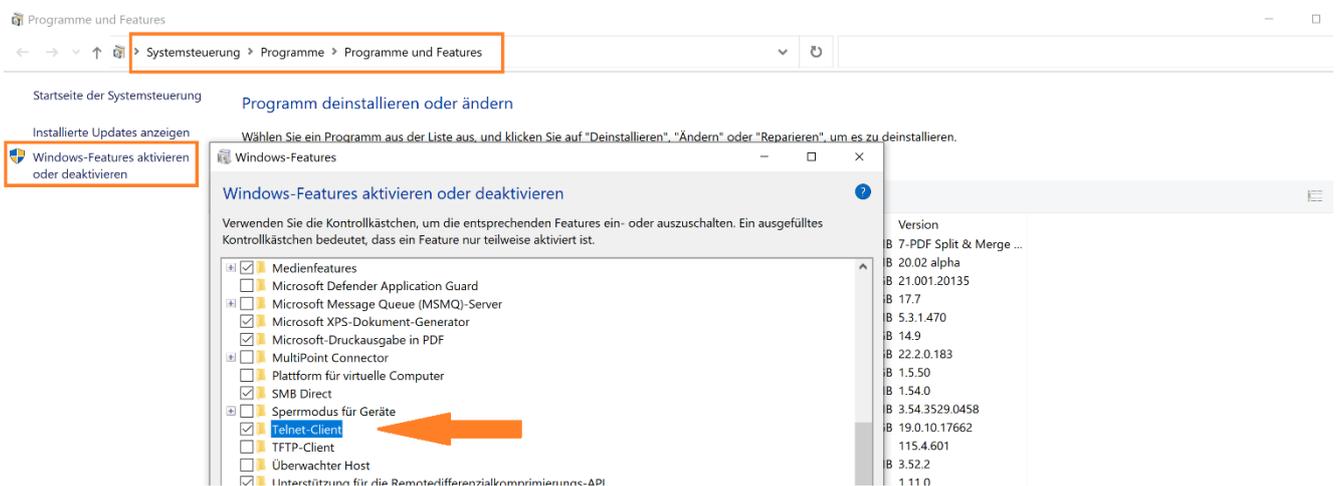


## Hands on HTTP

Den Datenaustausch im Hintergrund zwischen Web-Browser und -Server mittels dem «HyperText Transfer Protokoll» (HTTP) kann man selbst sehr einfach ausprobieren. Da HTTP **textbasiert** ist, kann man es von Hand nachvollziehen.

Wir werden den Web-Browser simulieren, indem wir Befehle zum Webserver schicken und die Antworten auf dem Bildschirm beobachten.

Dazu benötigt man lediglich den **Telnet-Client**, der den Zugriff auf entfernte Rechner erlaubt. Dieser muss allenfalls vorher unter Systemsteuerung/Programme/Programme und Features (dann Windows-Features aktivieren oder deaktivieren) aktiviert werden



### Vorgehen

Wenn die Telnet-Verbindung zum Webserver aufgebaut ist, wird alles, was man tippt, zum entfernten Rechner übertragen und alles, was der Rechner antwortet, erscheint auf dem Bildschirm.

Für den Aufbau der Verbindung sind zwei Angaben erforderlich:

- der Rechner-Name oder die **IP-Adresse**
- die **Port-Nummer**: Im Beispiel 80 (für den Webserver). Wenn keine Portnummer angegeben wird, verwendet der Browser standardmässig die Portnummer 80. (Achtung: Telnet verwendet aber standardmässig Port 23!)

### Abrufen und Darstellung der Datei im Browser

Tippen Sie im Webbrowser die Adresse des gewünschten Webserver ein. Verwenden Sie die Adresse, welche Ihnen die Lehrperson angibt.

Im Folgendem Ablauf wird der Server mit der Adresse 10.1.43.10 verwendet. Passen Sie diese IP-Adresse entsprechend an.



**Hallo beim Modul 239**

ich bin eine Web-Seite

**Die IP-Adresse oben kann sich ändern (Angaben gem. Lehrperson)**

Damit wird eine HTML-Datei (= Web-Seite) vom Server angefragt. Die enthaltenen textbasierten HTML-Anweisungen werden interpretiert und die Seite wird im Browser angezeigt.

Über Ansicht -> Seitenquelltext oder einen Editor ihrer Wahl können Sie den effektiv übertragenen HTML-Inhalt ansehen.

```
index_fuer_239.html
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2    "http://www.w3.org/TR/html4/loose.dtd">
3  <html>
4    <head>
5      <title>Hallo</title>
6    </head>
7    <body>
8      <center>
9        <h1>Hallo beim Modul 239</h1>
10       <h2><p />ich bin eine <b>Web-Seite</b></h2>
11     </center>
12   </body>
13 </html>
14
```



## Verbindungsaufbau mit "Telnet"

Telnet aufstarten ...

- Windows: „Start/Ausführen“ -> „telnet“ eingeben und <Enter> drücken.

```
Administrator: Eingabeaufforderung ←
C:\Windows\system32>telnet
```

Der Prompt „Microsoft Telnet>“ zeigt an, dass Telnet gestartet ist und auf Eingaben wartet:

```
Administrator: Eingabeaufforderung - telnet
Willkommen

Das Escapezeichen ist 'CTRL+'

Microsoft Telnet> _
```

TCP-Verbindung zum Server aufbauen (Achtung auf Port 80!):

**open 10.1.43.10 80** eintippen und mit der „Enter“-Taste bestätigen.

```
Telnet 10.1.43.10
Willkommen

Das Escapezeichen ist 'CTRL+'

Microsoft Telnet> open 10.1.43.10 80
Verbindungsaufbau zu 10.1.43.10...
```

Jetzt ist die TCP-Verbindung zum Port „80“ des Servers mittels Telnet aufgebaut.

Um zu überprüfen, ob die TCP-Verbindung auch wirklich aufgebaut wurde, kann man ein weiteres Admin-Terminal auf dem Client öffnen und darin das folgende Kommando absetzen:

**netstat -an**



Auf dem Bild unten ist ersichtlich, dass auf Layer 4 eine TCP-Verbindung hergestellt worden ist.

```
Eingabeaufforderung
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\marcello>netstat -an
Aktive Verbindungen

Proto Lokale Adresse Remoteadresse Status
TCP 0.0.0.0:135 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:443 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:445 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:903 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:913 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:5040 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:5357 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:7680 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49664 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49665 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49666 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49667 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49668 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:49670 0.0.0.0:0 ABHÖREN
TCP 0.0.0.0:54399 0.0.0.0:0 ABHÖREN
TCP 10.1.40.124:139 0.0.0.0:0 ABHÖREN
TCP 10.1.43.124:61430 10.1.43.10:80 HERGESTELLT
TCP 127.0.0.1:843 0.0.0.0:0 ABHÖREN
TCP 127.0.0.1:8307 0.0.0.0:0 ABHÖREN
TCP 127.0.0.1:15292 0.0.0.0:0 ABHÖREN
```

auf anderem Terminal, nach dem Befehl  
"open 10.1.43.10 80"

Der Webserver wartet jetzt auf eine Anfrage (einen Request). Sobald etwas eingetippt wird, springt der Cursor auf Ecke oben links.

Hinweis: Diese Schnittstelle wurde grundsätzlich für Rechner geschaffen. Dies hat verschiedene Konsequenzen:

- Die Eingaben **müssen** stimmen – Maschinen machen keine Schreibfehler. Wenn wir uns vertippen, müssen wir nochmals beginnen.
- Teilweise überlagern sich Eingaben und Ausgaben. Lassen Sie sich dadurch nicht beirren.
- Evtl. erhalten wir keine Rückmeldung zu unseren Eingaben – wir müssen «im Blindflug» arbeiten. Falls dem so ist, kann man in Telnet auch das lokale Echo einschalten.



## Anfordern einer Datei von Webserver

Eine Datei kann vom Server mit der HTTP-Methode **GET** angefordert werden.

Beispiel: Die Datei befindet sich auf dem Webserver im Root-Verzeichnis (/). Als Protokoll wird ‚HTTP/1.1‘ verwendet.

- **GET / HTTP/1.1** auf einer Zeile eintippen und mit „Enter“ auf eine neue Zeile wechseln. Tipp: Vor der Eingabe des GET-Befehls kann man zur besseren Übersicht die Enter-Taste so oft drücken, bis der Cursor auf einer freien Zeile steht.
- **Host: 10.1.43.10** eintippen und mit «Enter» auf eine neue Zeile wechseln.
- Das HTTP-Protokoll erwartet als Abschluss der Anfrage noch eine Leerzeile. Also „Enter“-Taste nochmals drücken (das darf auch nicht vergessen werden).

---

```
C:\> Telnet 10.1.43.10

Willkommen

Das Escapezeichen ist 'CTRL+'

Microsoft Telnet> open 10.1.43.10 80
Verbindungsaufbau zu 10.1.43.10...

GET / HTTP/1.1
Host: 10.1.43.10
```

Achtung: Es darf kein einziger Tippfehler gemacht werden. Das HTTP-Protokoll ist für die Kommunikation zwischen zwei Rechnern konzipiert worden, die keine Tippfehler machen.

Die Antwort des Webserver auf die GET-Anfrage enthält den **HTTP-Status**. Danach folgt der **HTTP-Header**. Dieser enthält Angaben zum Server und Informationen zur angeforderten Datei, wie Grösse, Änderungsdatum und Dateityp.

Nach einer Leerzeile folgt dann der **HTTP-Body** mit dem Datei-Inhalt. Ein Vergleich mit dem Quelltext im Browser zeigt, dass es sich um den gleichen Inhalt handelt.



```
Telnet 10.1.43.10

HTTP/1.1 200 OK ← Status-Code
Date: Sun, 14 Feb 2021 22:01:30 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Sun, 14 Feb 2021 19:48:59 GMT
ETag: "120-5bb5129179a41"
Accept-Ranges: bytes
Content-Length: 288
Vary: Accept-Encoding
Connection: close ← Nach Transaktion, Verbindung abbauen
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Hallo</title>
  </head>
  <body>
    <center>
      <h1>Hallo beim Modul 239</h1>
      <h2><p />ich bin eine <b>Web-Seite</b></h2>
    </center>
  </body>
</html>

TCP-Verbindung bleibt auf "SCHLIESSEN_WARTEN" bis ich mit beliebiger Taste bestätige
Verbindung zu Host verloren.
Drücken Sie eine beliebige Taste, um den Vorgang fortzusetzen..._
```

Die TCP-Verbindung zum Server wird nach der Auslieferung der Datei meist relativ schnell abgebaut. Manchmal reicht es jedoch noch für eine weitere Anfrage.

Das ist auch zu erkennen, wenn man auf der Admin-Konsole des Clients mit

`netstat -p tcp`

den entsprechenden Port überprüft (Siehe Bild unten). Da ist gut ersichtlich, dass dieser noch geöffnet ist und auf «SCHLIESSEN» wartet.

```
Eingabeaufforderung - netstat -p tcp

C:\Users\marcello>netstat -p tcp

Aktive Verbindungen

Proto Lokale Adresse Remoteadresse Status
TCP 10.1.40.124:53722 10.1.40.43:5240 HERGESTELLT
TCP 10.1.43.124:64436 10.1.43.10:http SCHLIESSEN_WARTEN
TCP 10.1.43.124:64488 10.1.43.10:ssh HERGESTELLT
```



## Bemerkung

- Wenn die erhaltene Webseite Verweise auf eingebettete Graphiken und Multimedia-Dokumente enthält, muss der Browser diese in nachfolgenden, separaten HTTP-Requests anfordern.
- Der manuelle Einsatz in diesem Beispiel soll das Prinzip des HTTP-Protokolls aufzeigen und ist nicht für den produktiven Einsatz gedacht.

## Zusätzliche Aufgaben

1. Wie sieht die Anfrage aus, wenn sie anstelle von HTTP 1.1 mit HTTP 2 gemacht werden soll?
2. Machen Sie weitere Anfragen mit anderen Methoden, z.B. mit **HEAD**, **OPTIONS**, ...
3. Setzen Sie mal auf der «Git Bash» oder einer Linux-Konsole das Kommando «curl» ein. Was macht dieses Kommando?
  - `curl --head «IP des Webservers»`
  - `curl «IP des Webservers»`