



HTTP - Hypertext Transfer Protocol

das Protokoll, um Daten im Web zu übertragen

Was ist HTTP?

HTTP ist das Kommunikationsprotokoll im World Wide Web (WWW). Die wichtigste Funktion ist, **Dateien vom Webserver anzufordern** und in den Browser zu laden.

Der Browser übernimmt dann die Darstellung von Texten und Bilder und kümmert sich um das Abspielen von Audio- und Video-Daten.

HTTP ist im OSI-Modell auf der Anwendungsebene beheimatet.

HTTP

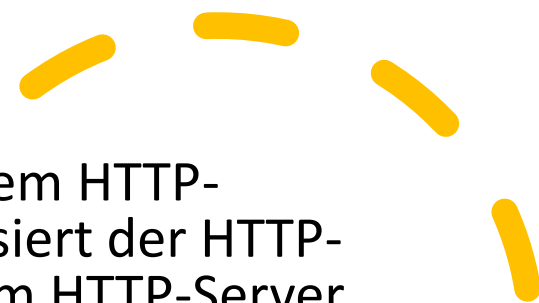
Die Kommunikation findet nach dem Client-Server-Prinzip statt. Der HTTP-Client (Browser) sendet seine Anfrage (HTTP-Request) an den HTTP-Server (Webserver).

Dieser bearbeitet die Anfrage und schickt seine Antwort (HTTP-Response) zurück. Nach der Antwort durch den Server ist diese Verbindung beendet. **Deshalb wird HTTP auch als zustandslos bezeichnet.**

Typischerweise finden gleichzeitig mehrere HTTP-Verbindungen statt.



HTTP- Adressierung



Damit der Server weiß, was er dem HTTP-Client zurückschicken soll, adressiert der HTTP-Client eine Datei, die sich auf dem HTTP-Server befinden muss.

Dazu wird vom HTTP-Client ein URL (Uniform Resource Locator) im HTTP-Header an den HTTP-Server übermittelt.

Beispiel einer URL:

<http://www.example.com>



HTTP- Request

Der HTTP-Request ist die Anfrage des HTTP-Clients an den HTTP-Server. Ein HTTP-Request besteht aus den Angaben **Methode**, URL und dem Request-Header.

Die häufigste **Methoden** sind **GET** und **POST**. Dahinter folgt durch ein Leerzeichen getrennt der URL und die verwendete HTTP-Version.

In weiteren Zeilen folgt der Header und bei der Methode POST durch eine Leerzeile (!) getrennt die Formular-Daten.

HTTP Request Header Beispiel

GET / HTTP/1.1

Host: www.example.de

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.1.2) Gecko/20090729 Firefox/3.5.2 (.NET CLR 3.5.30729)

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: de-de;q=0.8,en-us;q=0.5,en;q=0.3

Accept-Encoding: gzip,deflate

Accept-Charset: utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Token: 234802823048230

HTTP Methoden

Jeder HTTP-Request durch den Client wird durch die Angabe einer Methode eingeleitet. Die Methode teilt dem Server mit, was er mit dem Request machen soll.

GET:

Die GET-Methode dient der **Anforderung einer HTML-Datei**. Über GET lassen sich auch Formular-Daten übermitteln. Diese werden in codierter Form der URL angehängt. URL und Formular-Daten sind durch ein Fragezeichen (?) voneinander getrennt. Diese Parameter-String nennt sich Query-String und dient zum Übermitteln von zusätzlicher Information.

POST:

Dient in erster Linie zur Übertragung von Formulardaten an den Server.

es gibt noch weitere Methoden wie PUT, PATCH, TRACE, OPTIONS, HEAD, die je nach Server verfügbar sind.



URL in Browser


Wird eine HTTP-Adresse in die Browserleiste eingegeben, handelt es sich hierbei IMMER um eine Get-Anfrage. Es ist nicht möglich, von hier aus eine POST-Anfrage zu starten.

`https://example.com?show=True&id=3`

Get-Request an den Server example.com.



HTTP RESPONSE



Der HTTP-Response ist die Antwort des HTTP-Servers an den HTTP-Client. Der HTTP-Response besteht aus der verwendeten **HTTP-Version**, dem **Status-Code**, dem Inhalt, dem **HTTP Header** und dem **HTTP Statuscode**, zb. 200.

HTTP Response Header Beispiel

Beispiel für einen HTTP-Response-Header

HTTP/1.x 200 OK

Date: Tue, 08 Sep 2009 15:47:06 GMT

Server: Apache/1.3.34 Ben-SSL/1.55

Keep-Alive: timeout=2, max=200

Connection: Keep-Alive

Transfer-Encoding: chunked

Content-Type: text/html



https

HTTPS ist eine erweiterte Form des HTTP.

Es ist eine Erweiterung von HTTP. Das "S" in der Abkürzung stammt vom Wort 'secure' und es wird von Transport Layer Security (TLS) [der Nachfolger von [Secure Sockets Layer \(SSL\)](#)] unterstützt, der Standard-Sicherheitstechnologie, die eine verschlüsselte Verbindung zwischen einem Webserver und einem Browser herstellt.

Nachrichten zwischen Client und Server werden also verschlüsselt übertragen (Public-Key-Verfahren).

Versionen

HTTP/1.1

1997 erschien die Version HTTP/1.1, beschrieben im Memo RFC2068. Sie galt als erster „offizieller“ Standard und ist **bis heute in Gebrauch**.

HTTP/2

Im Laufe der Jahre wurden die Websites immer größer und komplexer. Um eine moderne Website in den Browser zu laden, muss dieser mehrere Megabyte Daten anfordern und bis zu hundert einzelne HTTP-Requests absenden. Da HTTP/1.1 vorsieht, dass die Requests innerhalb einer Verbindung nacheinander abgearbeitet werden, dauert der Seitenaufbau umso länger, je komplexer eine Webseite ist. http/2 arbeitet parallel.

HTTP/3

nutzt UDP (QUIC) statt TCP, da der TCP-Handshake zu lange dauert. Immer noch in der Entwicklung. Wird von Google in experimentellen Protokollen genutzt,