HTTP

+1

Published Apr 5, 2022•**Updated Oct 26, 2022**

Contribute to Docs

Hypertext Transfer Protocol (HTTP) is used for fetching HTML documents and other web-based resources. It follows a model where a client (most commonly a browser) requests content stored on a server.

HTTPS

Instead of HTTP, a more secure version, HTTPS, should be used in order to encrypt information sent between the client and server. This encryption is done with Transport Layer Security (or TLS, formerly SSL). Some benefits of HTTPS include:

- Financial data, like credit card information or bank account numbers, are protected from interception.
- Domain-ownership can be better verified by users.
- A growing standard of trust around sites that use HTTPS.

Websites can be certified with HTTPS so that browsers "know" the official site for a person, business, etc. These certifications are approved and signed by a trusted certificate authority (CA).

Requests

HTTP requests are stateless, meaning that all requests are independent and have no knowledge of one another. Requests contain the following parts:

- The HTTP method being used (more information shortly).
- The requested URL along with any queries or parameters.
- The HTTP version, such as 1.1 or 2.0.
- Any header information such as:
    - The referer that tells the URL where the request came from.
    - Any user agent information about the requesting client.
    - A unique host name that is ideal for many pages on one server.
    - Cookie data about the request.
- A response-like body that contains the resource to be sent (common with the POST method).

An HTTP client requests information specific URLs using four primary methods:

Method	Description
GET	Requests data, content, or other resources from the server.
POST	Sends data, content, or other resources to the server.
PUT	Sends updates for existing content on the server.
DELETE	Deletes specific content from the server.

Responses

If the server is able to connect with the client and fulfill its request, it will send back a response that includes the following parts:

- The version of HTTP being used.
- Headers similar to the ones used for HTTP requests.
- A body that contains the successfully requested resource.
- A status code with a message explaining why the request succeeded or failed.

A breakdown of response status codes is shown below.

Informational

Status Code	Name	Description
100	Continue	The request should continue or be ignored if finished.
102	Processing	The server is currently processing the request; no response yet

Successful

Status Code	Name	Description
200	OK	The request is successful and a response was sent.
202	Accepted	Processing not yet finished but request was accepted.

Redirection

Status Code	Name	Description
301	Moved Permanently	The resource URL was changed and the new one was sent in t response.
304	Not Modified	Used by caches for serving the same, unmodified content.

Client Errors

Status Code	Name	Description
400	Bad Request	Invalid request based on client-side error (invalid syntax, URL, etc.).
401	Unauthorized	Invalid client credentials, such as an API key.
404	Not Found	Server couldn’t find resource (e.g. invalid URL).
408	Request Timeout	The request is taking too long to finish.

Server Errors

Status Code	Name	Description
500	Internal Server Error	Error occurred on the server-side.
502	Bad Gateway	Invalid response from a gateway or proxy server.

Caches and Proxies

HTTP can be used for improving web performance with caches and proxy servers.

Client browsers can use caches for serving content instead of making repeated requests for the same content. Examples of caches include:

- CDNs that retain copies of web content and serve from close network connections.
- Proxy browser caches, like ones used in progressive web apps that allow a single user to cache and access content offline.
- Shared proxy caches that store resources for multiple users (internet service provider, company staff network, etc.).

Proxies are used to mask a client’s IP address by assigning one to a proxy server and have requests sent from there.

All contributors

1. 

   

2.