

# Hypertext Transfer Protocol

[An overview of HTTP - HTTP | MDN \(mozilla.org\)](#)

[Hypertext Transfer Protocol – Wikipedia](#)

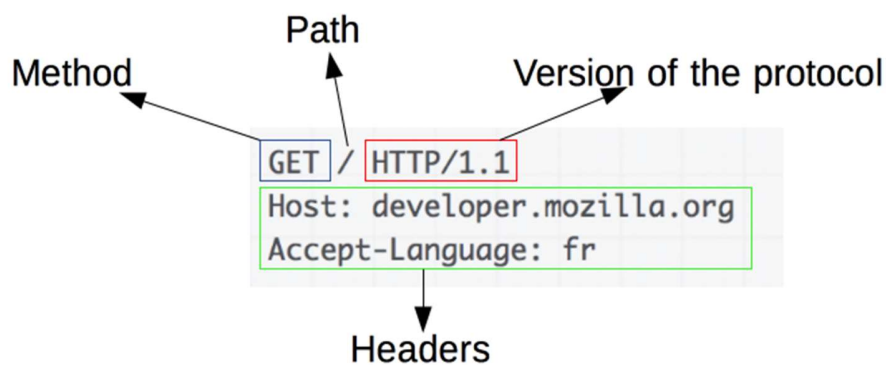
<https://www.ionos.at/digitalguide/hosting/hosting-technik/was-ist-http/>

## Was es ist und wo es genutzt wird.

- Client-Server-Protokoll
- Ursprünglich nur für Websites
- REST, Datenbankzugriff, Medienplayer, ...
- Zustandslos, Verbindungslos (bis HTTP/1), Medienunabhängig

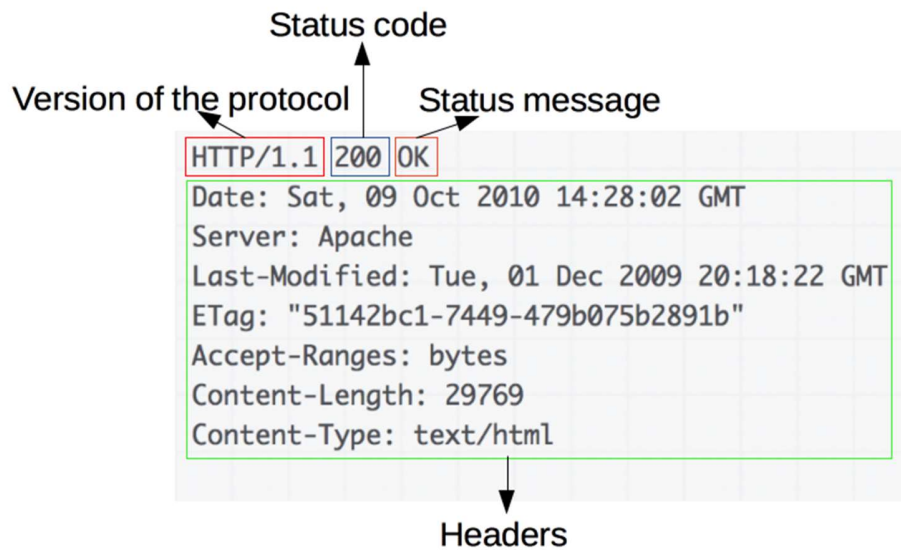
## HTTP Protokol Bestandteile

### Requests:



- HTTP Methode (GET,POST,...)
- Ressourcenpfad
- Version des Protokolls
- Optionaler Nachrichtenkopf (Message Header)
- Optionaler Nachrichtenrumpf (Message Body)

### Responses:



- Version des Protokolls
- Statuscode der Serverantwort
- Statusmessage des Servers
- Optionaler Nachrichtenkopf (Message Header)
- Optionaler Nachrichtenrumpf (Message Body)

#### HTTP Anfragemethoden

- GET
- POST
- HEAD
- PUT
- PATCH
- DELETE
- TRACE
- OPTIONS
- CONNECT

#### HTTP Status Codes

1xx informational response – the request was received, continuing process

2xx successful – the request was successfully received, understood, and accepted

3xx redirection – further action needs to be taken in order to complete the request

4xx client error – the request contains bad syntax or cannot be fulfilled

5xx server error – the server failed to fulfil an apparently valid request

#### 200 OK

This is the standard “OK” status code for a successful HTTP request. The response that is returned is dependent on the request. For example, for a GET request, the response will be included in the message body. For a PUT/POST request, the response will include the resource that contains the result of the action.

#### 201 Created

This is the status code that confirms that the request was successful and, as a result, a new resource was created. Typically, this is the status code that is sent after a POST/PUT request.

#### 204 No Content

This status code confirms that the server has fulfilled the request but does not need to return information.

#### 304 Not Modified

This status code is used for browser caching. If the response has not been modified, the client/user can continue to use the same response/cached version.

#### 400 Bad Request

The server cannot understand and process a request due to a client error. Missing data, domain validation, and invalid formatting are some examples that cause the status code 400 to be sent.

#### 401 Unauthorized (RFC 7235)

This status code request occurs when authentication is required but has failed or not been provided.

#### 403 Forbidden

Very similar to status code 401, a status code 403 happens when a valid request was sent, but the server refuses to accept it. This happens if a client/user requires the necessary permission or they may need an account to access the resource. Unlike a status code 401, authentication will not apply here.

#### 404 Not Found

The most common status code the average user will see. A status code 404 occurs when the request is valid, but the resource cannot be found on the server. Even though

these are grouped in the Client Errors “bucket,” they are often due to improper URL redirection.

#### **409 Conflict**

A status code 409 is sent when a request conflicts with the current state of the resource. This is usually an issue with simultaneous updates, or versions, that conflict with one another.

#### **410 Gone**

Resource requested is no longer available and will not be available again.

#### **500 Internal Server Error**

Another one of the more commonly seen status codes by users, the 500 series codes are similar to the 400 series codes in that they are true error codes. The status code 500 happens when the server cannot fulfill a request due to an unexpected issue. Web developers typically have to comb through the server logs to determine where the exact issue is coming from.