HTTP Methods and Codes

HTTP supports multiple methods for accessing a resource. In the HTTP protocol, several request methods allow the browser to send information, forms, or files to the server. These methods are used, among other things, to tell the server how to process the request we send and how to reply.

We saw different HTTP methods used in the HTTP requests we tested in the previous sections. With cURL, if we use `-v` to preview the full request, the first line contains the HTTP method (e.g. `GET / HTTP/1.1`), while with browser devtools, the HTTP method is shown in the `Method` column. Furthermore, the response headers also contain the HTTP response code, which states the status of processing our HTTP request.

Request Methods

The following are some of the commonly used methods:

Method	Description
GET	Requests a specific resource. Additional data can be passed to the server via query strings in the URL (e.g. `?param=value`).
POST	Sends data to the server. It can handle multiple types of input, such as text, PDFs, and other forms of binary data. This data is appended in the request body present after the headers. The POST method is commonly used when sending information (e.g. forms/logins) or uploading data to a website, such as images or documents.
HEAD	Requests the headers that would be returned if a GET request was made to the server. It doesn't return the request body and is usually made to check the response length before downloading resources.
PUT	Creates new resources on the server. Allowing this method without proper controls can lead to uploading malicious resources.
DELETE	Deletes an existing resource on the webserver. If not properly secured, can lead to Denial of Service (DoS) by deleting critical files on the web server.
OPTIONS	Returns information about the server, such as the methods accepted by it.
PATCH	Applies partial modifications to the resource at the specified location.

The list only highlights a few of the most commonly used HTTP methods. The availability of a particular method depends on the server as well as the application configuration. For a full list of HTTP methods, you can visit this link.

Note: Most modern web applications mainly rely on the `GET` and `POST` methods. However, any web application that utilizes REST APIs also rely on `PUT` and `DELETE`, which are used to update and delete data on the API endpoint, respectively. Refer to the Introduction to Web Applications module for more details.

Response Codes

HTTP status codes are used to tell the client the status of their request. An HTTP server can return five types of response codes:

Type	Description
1xx	Provides information and does not affect the processing of the request.
2xx	Returned when a request succeeds.
3xx	Returned when the server redirects the client.
4xx	Signifies improper requests `from the client`. For example, requesting a resource that doesn't exist or requesting a bad format.

Type	Description
5xx	Returned when there is some problem `with the HTTP server` itself.

The following are some of the commonly seen examples from each of the above HTTP method types:

Code	Description
200 OK	Returned on a successful request, and the response body usually contains the requested resource.
302 Found	Redirects the client to another URL. For example, redirecting the user to their dashboard after a successful login.
400 Bad Request	Returned on encountering malformed requests such as requests with missing line terminators.
403 Forbidden	Signifies that the client doesn't have appropriate access to the resource. It can also be returned when the server detects malicious input from the user.
404 Not Found	Returned when the client requests a resource that doesn't exist on the server.
500 Internal Server Error	Returned when the server cannot process the request.

For a a full list of standard HTTP response codes, you can visit this link. Apart from the standard HTTP codes, various servers and providers such as Cloudflare or AWS implement their own codes.

← Previous   Next →   ✓ Mark Complete & Next

📄 Cheat Sheet

OFFLINE

▶ Start Instance

1 / 1 spawns left