

Jonathan Klinger

5BHIF

Einfacher HTTP 1.1 Client

A short introduction to HTTP 1.1, it's requests
and a simple Client written in C++

Abteilung Informatik
HTBLuVA Wiener Neustadt
Jänner 2021

Contents

1	HTTP	2
1.1	HTTP 1.1	2
1.2	HTTP Requests	2
1.3	HTTP Authentication	2
1.4	HTTP Status Codes	2
2	My Solution	3
2.1	Difficulties	3
3	Sources	3

1 HTTP

The Hypertext Transfer Protocol is a protocol designed for transmitting hypermedia documents (e.g., HTML), which is settled in the application-layer of the OSI and TCP/IP model. The main focus of the protocol is on the communication between a client and a server.

HTTP is a stateless protocol.

Stateless: A protocol is stateless when it is not required for the Server to retain session information or a status about each communicating partner.

Usually clients send requests to the server which responds according to it.

1.1 HTTP 1.1

This version of the protocol was released in 1997 and came with a number of improvements to its predecessor HTTP 1.0 and 0.9. E.g., Chunked responses were now supported which means that the sender could add additional header after the message body.

1.2 HTTP Requests

HTTP comes with a number of possible requests. The type is specified in the header of the message. Here are the most important, which are also the same I implemented in my program:

- GET - request data from a specified source. This type of request can be cached and also be bookmarked.
- POST - send data to a server to create a resource. These are never cached and cannot be bookmarked.
- PUT - send data to a server to update or create a resource. Calling the same PUT request multiple times will always end in the same result while calling the same POST request multiple times will end in creating the same resource multiple times.
- DELETE - delete a specified resource.

1.3 HTTP Authentication

HTTP provides basic authentication which is implemented in the program. The server identifies issues before serving the requested content.

1.4 HTTP Status Codes

Every HTTP response comes with a status codes. There are enough to suit every situation possible. Here are the most important.

- 2xx - Success. Representative 200: OK
- 4xx - Client Errors. Representative 401: Unauthorized
- 5xx - Server Errors. Representative 500: Internal Server Error

With the help of these codes it is possible for a Client to react to the Server and vice versa.

2 My Solution

The language I coded my solution in is C++. It offers a lot of possible approaches. For example I decided to work with the C++ library `cpp-httplib` (linked below).

This library offers function for all the request-types that are explained above. With the help of `CLI11` and `spdlog` I created a program which makes it possible to send 3 requests and save the result to a file.

The program informs the user about every step taken. Starting with how the input is interpreted to a info that the response is successfully written to a file.

Error Messages from the Server are directly passed to the user as are the HTTP Status Codes.

2.1 Difficulties

The only difficulty that was encountered during the development was the conversion from string to char *. The input is saved in a vector of strings but the library needs char * passed on their functions. At first a function was made that assigned the reference to a string object to a char *. Later the function `"c_str()"` was found which converts strings to constant char *. Therefore all char * where made constant. Problem solved.

3 Sources

- https://www.w3schools.com/tags/ref_httpmethods.aspx
- https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP
- <https://www.w3.org/Protocols/rfc2616/rfc2616.html>
- <https://tools.ietf.org/html/rfc2616>
- <https://github.com/yhirose/cpp-httplib>