

NAME

CURLOPT_HTTP_VERSION – HTTP protocol version to use

SYNOPSIS

```
#include <curl/curl.h>
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HTTP_VERSION, long version);
```

DESCRIPTION

Pass *version* a long, set to one of the values described below. They ask libcurl to use the specific HTTP versions.

Note that the HTTP version is a request. libcurl still prioritizes to reuse existing connections so it might then reuse a connection using an HTTP version you have not asked for.

CURL_HTTP_VERSION_NONE

We do not care about what version the library uses. libcurl uses whatever it thinks fit.

CURL_HTTP_VERSION_1_0

Enforce HTTP 1.0 requests.

CURL_HTTP_VERSION_1_1

Enforce HTTP 1.1 requests.

CURL_HTTP_VERSION_2_0

Attempt HTTP 2 requests. libcurl falls back to HTTP 1.1 if HTTP 2 cannot be negotiated with the server.

When libcurl uses HTTP/2 over HTTPS, it does not itself insist on TLS 1.2 or higher even though that is required by the specification. A user can add this version requirement with *CURLOPT_SSLVERSION(3)*.

The alias *CURL_HTTP_VERSION_2* was added to better reflect the actual protocol name.

CURL_HTTP_VERSION_2TLS

Attempt HTTP 2 over TLS (HTTPS) only. libcurl falls back to HTTP 1.1 if HTTP 2 cannot be negotiated with the HTTPS server. For clear text HTTP servers, libcurl uses 1.1.

CURL_HTTP_VERSION_2_PRIOR_KNOWLEDGE

Issue non-TLS HTTP requests using HTTP/2 without HTTP/1.1 Upgrade. It requires prior knowledge that the server supports HTTP/2 straight away. HTTPS requests still do HTTP/2 the standard way with negotiated protocol version in the TLS handshake.

Since 8.10.0 if this option is set for an HTTPS request then the application layer protocol version (ALPN) offered to the server is only HTTP/2. Prior to that both HTTP/1.1 and HTTP/2 were offered.

CURL_HTTP_VERSION_3

(Added in 7.66.0) This option makes libcurl attempt to use HTTP/3 to the host given in the URL, with fallback to earlier HTTP versions if needed.

CURL_HTTP_VERSION_3ONLY

(Added in 7.88.0) Setting this makes libcurl attempt to use HTTP/3 directly to server given in the URL and does not downgrade to earlier HTTP version if the server does not support HTTP/3.

DEFAULT

Since curl 8.13.0: **CURL_HTTP_VERSION_NONE**

Since curl 7.62.0: **CURL_HTTP_VERSION_2TLS**

Before that: **CURL_HTTP_VERSION_1_1**

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        curl_easy_setopt(curl, CURLOPT_HTTP_VERSION, CURL_HTTP_VERSION_2TLS);
        result = curl_easy_perform(curl);
        if(result == CURLE_HTTP_RETURNED_ERROR) {
            /* an HTTP response error problem */
        }
    }
}
```

HISTORY

CURL_HTTP_VERSION_* enums became *long* types in 8.13.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.9.1

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURLE_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO

CURLOPT_ALTSVC(3), **CURLOPT_HTTP09_ALLOWED(3)**, **CURLOPT_HTTP200ALIASES(3)**, **CURLOPT_SSLVERSION(3)**