

**NAME**

CURLOPT\_HTTP\_VERSION – specify 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. This is not sensible to do unless you have a good reason. You have to set this option if you want to use libcurl's HTTP/2 support.

Note that the HTTP version is just a request. libcurl will still prioritize to re-use an existing connection so it might then re-use a connection using a HTTP version you haven't asked for.

CURL\_HTTP\_VERSION\_NONE

We don't care about what version the library uses. libcurl will use 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 will fall back to HTTP 1.1 if HTTP 2 can't be negotiated with the server. (Added in 7.33.0)

The alias *CURL\_HTTP\_VERSION\_2* was added in 7.43.0 to better reflect the actual protocol name.

CURL\_HTTP\_VERSION\_2TLS

Attempt HTTP 2 over TLS (HTTPS) only. libcurl will fall back to HTTP 1.1 if HTTP 2 can't be negotiated with the HTTPS server. For clear text HTTP servers, libcurl will use 1.1. (Added in 7.47.0)

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 will still do HTTP/2 the standard way with negotiated protocol version in the TLS handshake. (Added in 7.49.0)

**DEFAULT**

CURL\_HTTP\_VERSION\_NONE

**PROTOCOLS**

HTTP

**EXAMPLE**

TODO

**AVAILABILITY**

Along with HTTP

**RETURN VALUE**

Returns CURLE\_OK if HTTP is supported, and CURLE\_UNKNOWN\_OPTION if not.

**SEE ALSO**

CURLOPT\_SSLVERSION(3), CURLOPT\_HTTP200ALIASES(3),