NAME

CURLOPT_HTTPHEADER − set of HTTP headers

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HTTPHEADER,
                          struct curl_slist *headers);
```

DESCRIPTION

Pass a pointer to a linked list of HTTP headers to pass to the server and/or proxy in your HTTP request. The same list can be used for both host and proxy requests.

When used within an IMAP or SMTP request to upload a MIME mail, the given header list establishes the document−level MIME headers to prepend to the uploaded document described by *CURLOPT_MIMEPOST(3)*. This does not affect raw mail uploads.

When used with HTTP, this option can add new headers, replace internal headers and remove internal headers.

The linked list should be a valid list of **struct curl_slist** entries properly filled in. Use *curl_slist_append(3)* to create the list and *curl_slist_free_all(3)* to free it again after use.

If you provide a header that is otherwise generated and used by libcurl internally, your header alternative is used instead. If you provide a header without content (no data on the right side of the colon) as in *Accept:*, the internally used header is removed. To forcibly add a header without content (nothing after the colon), use the form *name;* (using a trailing semicolon).

There are exceptions when suppressing headers. The *Connection:* header in HTTP/1.1 cannot be overridden. You can provide values for it, but should a request require specific ones, they are always added to your own.

The headers included in the linked list **must not** be CRLF−terminated, since libcurl adds CRLF after each header item itself. Failure to comply with this might result in strange behavior. libcurl passes on the verbatim strings you give it, without any filter or other safe guards. That includes white space and control characters.

The first line in an HTTP request (containing the method, usually a GET or POST) is not a header and cannot be replaced using this option. Only the lines following the request−line are headers. Adding this method line in this list of headers only causes your request to send an invalid header. Use *CURLOPT_CUSTOMREQUEST(3)* to change the method.

When this option is passed to *curl_easy_setopt(3)*, libcurl does not copy the entire list so you **must** keep it around until you no longer use this *handle* for a transfer before you call *curl_slist_free_all(3)* on the list.

Using this option multiple times makes the last set list override the previous ones. Set it to NULL to disable its use again.

The most commonly replaced HTTP headers have "shortcuts" in the options *CURLOPT_COOKIE(3)*, *CURLOPT_USERAGENT(3)* and *CURLOPT_REFERER(3)*. We recommend using those.

There is an alternative option that sets or replaces headers only for requests that are sent with CONNECT to a proxy: *CURLOPT_PROXYHEADER(3)*. Use *CURLOPT_HEADEROPT(3)* to control the behavior.

SPECIFIC HTTP HEADERS

Setting some specific headers causes libcurl to act differently.

Host: The specified hostname is used for cookie matching if the cookie engine is also enabled for this transfer. If the request is done over HTTP/2 or HTTP/3, the custom hostname is instead used in the ":authority" header field and Host: is not sent at all over the wire.

Transfer-Encoding: chunked
: Tells libcurl the upload is to be done using this chunked encoding instead of providing the Content−Length: field in the request.

SPECIFIC MIME HEADERS

When used to build a MIME email for IMAP or SMTP, the following document−level headers can be set to override libcurl−generated values:

Mime-Version:
: Tells the parser at the receiving site how to interpret the MIME framing. It defaults to "1.0" and should normally not be altered.

Content-Type:
: Indicates the document's global structure type. By default, libcurl sets it to "multipart/mixed", describing a document made of independent parts. When a MIME mail is only composed of alternative representations of the same data (i.e.: HTML and plain text), this header must be set to "multipart/alternative". In all cases the value must be of the form "multipart/*" to respect the document structure and may not include the "boundary=" parameter.

Other specific headers that do not have a libcurl default value but are strongly desired by mail delivery and user agents should also be included. These are *From:*, *To:*, *Date:* and *Subject:* among others and their presence and value is generally checked by anti−spam utilities.

SECURITY CONCERNS

By default, this option makes libcurl send the given headers in all HTTP requests done by this handle. You should therefore use this option with caution if you for example connect to the remote site using a proxy and a CONNECT request, you should to consider if that proxy is supposed to also get the headers. They may be private or otherwise sensitive to leak.

Use *CURLOPT_HEADEROPT(3)* to make the headers only get sent to where you intend them to get sent.

Custom headers are sent in all requests done by the easy handle, which implies that if you tell libcurl to follow redirects (*CURLOPT_FOLLOWLOCATION(3)*), the same set of custom headers is sent in the subsequent request. Redirects can of course go to other hosts and thus those servers get all the contents of your custom headers too.

Starting in 7.58.0, libcurl specifically prevents "Authorization:" headers from being sent to other hosts than the first used one, unless specifically permitted with the *CURLOPT_UNRESTRICTED_AUTH(3)* option.

Starting in 7.64.0, libcurl specifically prevents "Cookie:" headers from being sent to other hosts than the first used one, unless specifically permitted with the *CURLOPT_UNRESTRICTED_AUTH(3)* option.

DEFAULT

NULL

PROTOCOLS

This functionality affects http, imap and smtp

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
```

```
 struct curl_slist *list = NULL;

 if(curl) {
  CURLcode result;
  curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

  /* add this header */
  list = curl_slist_append(list, "Shoesize: 10");

  /* remove this header */
  list = curl_slist_append(list, "Accept:");

  /* change this header */
  list = curl_slist_append(list, "Host: example.net");

  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, list);

  result = curl_easy_perform(curl);

  curl_slist_free_all(list); /* free the list */
  curl_easy_cleanup(curl);
 }
}
```

HISTORY

Use for MIME mail added in 7.56.0.

AVAILABILITY

Added in curl 7.1

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_CUSTOMREQUEST(3), **CURLOPT_HEADER**(3), **CURLOPT_HEADEROPT**(3), **CURLOPT_MIMEPOST**(3), **CURLOPT_PROXYHEADER**(3), **curl_mime_init**(3)