

NAME

CURLOPT_HEADER – pass headers to the data stream

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HEADER, long onoff);
```

DESCRIPTION

Pass the long value *onoff* set to 1 to ask libcurl to include the headers in the write callback (*CURLOPT_WRITEFUNCTION(3)*). This option is relevant for protocols that actually have headers or other meta-data (like HTTP and FTP).

When asking to get the headers passed to the same callback as the body, it is not possible to accurately separate them again without detailed knowledge about the protocol in use.

Further: the *CURLOPT_WRITEFUNCTION(3)* callback is limited to only ever get a maximum of *CURL_MAX_WRITE_SIZE* bytes passed to it (16KB), while a header can be longer and the *CURLOPT_HEADERFUNCTION(3)* supports getting called with headers up to *CURL_MAX_HTTP_HEADER* bytes big (100KB).

It is often better to use *CURLOPT_HEADERFUNCTION(3)* to get the header data separately.

While named confusingly similar, *CURLOPT_HTTPHEADER(3)* is used to set custom HTTP headers.

DEFAULT

0

PROTOCOLS

This functionality affects ftp, http, imap, pop3 and smtp

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_HEADER, 1L);

        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

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_HEADERFUNCTION(3), *CURLOPT_HTTPHEADER(3)*