

NAME

`curl_easy_getinfo` - extract information from a curl handle

SYNOPSIS

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

```
CURLcode curl_easy_getinfo(CURL *curl, CURLINFO info, ... );
```

DESCRIPTION

Request internal information from the curl session with this function. The third argument **MUST** be a pointer to a long, a pointer to a char *, a pointer to a struct `curl_slist *` or a pointer to a double (as this documentation describes further down). The data pointed-to will be filled in accordingly and can be relied upon only if the function returns `CURLE_OK`. Use this function **AFTER** a performed transfer if you want to get transfer- oriented data.

You should not free the memory returned by this function unless it is explicitly mentioned below.

AVAILABLE INFORMATION

The following information can be extracted:

CURLINFO_EFFECTIVE_URL

Pass a pointer to a 'char *' to receive the last used effective URL.

CURLINFO_RESPONSE_CODE

Pass a pointer to a long to receive the last received HTTP or FTP code. This option was known as `CURLINFO_HTTP_CODE` in libcurl 7.10.7 and earlier. This will be zero if no server response code has been received. Note that a proxy's `CONNECT` response should be read with `CURLINFO_HTTP_CONNECTCODE` and not this.

CURLINFO_HTTP_CONNECTCODE

Pass a pointer to a long to receive the last received proxy response code to a `CONNECT` request.

CURLINFO_FILETIME

Pass a pointer to a long to receive the remote time of the retrieved document (in number of seconds since 1 jan 1970 in the GMT/UTC time zone). If you get -1, it can be because of many reasons (unknown, the server hides it or the server doesn't support the command that tells document time etc) and the time of the document is unknown. Note that you must tell the server to collect this information before the transfer is made, by using the `CURLOPT_FILETIME` option to `curl_easy_setopt(3)` or you will unconditionally get a -1 back. (Added in 7.5)

CURLINFO_TOTAL_TIME

Pass a pointer to a double to receive the total transaction time in seconds for the previous transfer. This time does not include the connect time, so if you want the complete operation time, you should add the `CURLINFO_CONNECT_TIME`.

CURLINFO_NAMELOOKUP_TIME

Pass a pointer to a double to receive the time, in seconds, it took from the start until the name resolving was completed.

CURLINFO_CONNECT_TIME

Pass a pointer to a double to receive the time, in seconds, it took from the start until the connect to the remote host (or proxy) was completed.

CURLINFO_PRETRANSFER_TIME

Pass a pointer to a double to receive the time, in seconds, it took from the start until the file transfer is just about to begin. This includes all pre-transfer commands and negotiations that are specific to the particular protocol(s) involved.

CURLINFO_STARTTRANSFER_TIME

Pass a pointer to a double to receive the time, in seconds, it took from the start until the first byte is just about to be transferred. This includes `CURLINFO_PRETRANSFER_TIME` and also the time

the server needs to calculate the result.

CURLINFO_REDIRECT_TIME

Pass a pointer to a double to receive the total time, in seconds, it took for all redirection steps include name lookup, connect, pretransfer and transfer before final transaction was started. CURLINFO_REDIRECT_TIME contains the complete execution time for multiple redirections. (Added in 7.9.7)

CURLINFO_REDIRECT_COUNT

Pass a pointer to a long to receive the total number of redirections that were actually followed. (Added in 7.9.7)

CURLINFO_SIZE_UPLOAD

Pass a pointer to a double to receive the total amount of bytes that were uploaded.

CURLINFO_SIZE_DOWNLOAD

Pass a pointer to a double to receive the total amount of bytes that were downloaded. The amount is only for the latest transfer and will be reset again for each new transfer.

CURLINFO_SPEED_DOWNLOAD

Pass a pointer to a double to receive the average download speed that curl measured for the complete download.

CURLINFO_SPEED_UPLOAD

Pass a pointer to a double to receive the average upload speed that curl measured for the complete upload.

CURLINFO_HEADER_SIZE

Pass a pointer to a long to receive the total size of all the headers received.

CURLINFO_REQUEST_SIZE

Pass a pointer to a long to receive the total size of the issued requests. This is so far only for HTTP requests. Note that this may be more than one request if FOLLOWLOCATION is true.

CURLINFO_SSL_VERIFYRESULT

Pass a pointer to a long to receive the result of the certification verification that was requested (using the CURLOPT_SSL_VERIFYPEER option to *curl_easy_setopt(3)*).

CURLINFO_SSL_ENGINES

Pass the address of a 'struct curl_slist *' to receive a linked-list of OpenSSL crypto-engines supported. Note that engines are normally implemented in separate dynamic libraries. Hence not all the returned engines may be available at run-time. **NOTE:** you must call *curl_slist_free_all(3)* on the list pointer once you're done with it, as libcurl will not free the data for you. (Added in 7.12.3)

CURLINFO_CONTENT_LENGTH_DOWNLOAD

Pass a pointer to a double to receive the content-length of the download. This is the value read from the Content-Length: field.

CURLINFO_CONTENT_LENGTH_UPLOAD

Pass a pointer to a double to receive the specified size of the upload.

CURLINFO_CONTENT_TYPE

Pass a pointer to a 'char *' to receive the content-type of the downloaded object. This is the value read from the Content-Type: field. If you get NULL, it means that the server didn't send a valid Content-Type header or that the protocol used doesn't support this.

CURLINFO_PRIVATE

Pass a pointer to a 'char *' to receive the pointer to the private data associated with the curl handle (set with the CURLOPT_PRIVATE option to *curl_easy_setopt(3)*). (Added in 7.10.3)

CURLINFO_HTTPAUTH_AVAIL

Pass a pointer to a long to receive a bitmask indicating the authentication method(s) available. The meaning of the bits is explained in the CURLOPT_HTTPAUTH option for *curl_easy_setopt(3)*.

(Added in 7.10.8)

CURLINFO_PROXYAUTH_AVAIL

Pass a pointer to a long to receive a bitmask indicating the authentication method(s) available for your proxy authentication. (Added in 7.10.8)

CURLINFO_OS_ERRNO

Pass a pointer to a long to receive the errno variable from a connect failure. (Added in 7.12.2)

CURLINFO_NUM_CONNECTS

Pass a pointer to a long to receive how many new connections libcurl had to create to achieve the previous transfer (only the successful connects are counted). Combined with *CURLINFO_REDIRECT_COUNT* you are able to know how many times libcurl successfully reused existing connection(s) or not. See the Connection Options of *curl_easy_setopt(3)* to see how libcurl tries to make persistent connections to save time. (Added in 7.12.3)

CURLINFO_COOKIELIST

Pass a pointer to a 'struct curl_slist *' to receive a linked-list of all cookies cURL knows (expired ones, too). Don't forget to *curl_slist_free_all(3)* the list after it has been used. If there are no cookies (cookies for the handle have not been enabled or simply none have been received) 'struct curl_slist *' will be set to point to NULL. (Added in 7.14.1)

CURLINFO_LASTSOCKET

Pass a pointer to a long to receive the last socket used by this curl session. If the socket is no longer valid, -1 is returned. When you finish working with the socket, you must call *curl_easy_cleanup()* as usual and let libcurl close the socket and cleanup other resources associated with the handle. This is typically used in combination with *CURLOPT_CONNECT_ONLY*. (Added in 7.15.2)

TIMES

An overview of the six time values available from *curl_easy_getinfo()*

curl_easy_perform()

```
|
|--NT
|--CT
|--PT
|--ST
|  |--TT
|--RT
```

- NT *CURLINFO_NAMELOOKUP_TIME*. The time it took from the start until the name resolving was completed.
- CT *CURLINFO_CONNECT_TIME*. The time it took from the start until the connect to the remote host (or proxy) was completed.
- PT *CURLINFO_PRETRANSFER_TIME*. The time it took from the start until the file transfer is just about to begin. This includes all pre-transfer commands and negotiations that are specific to the particular protocol(s) involved.
- ST *CURLINFO_STARTTRANSFER_TIME*. The time it took from the start until the first byte is just about to be transferred.
- TT *CURLINFO_TOTAL_TIME*. Time of the previous transfer. This time does not include the connect time (CT), so if you want the complete operation time, you should add that.
- RT *CURLINFO_REDIRECT_TIME*. The time it took for all redirection steps include name lookup, connect, pretransfer and transfer before final transaction was started. So, this is zero if no redirection took place.

RETURN VALUE

If the operation was successful, CURLE_OK is returned. Otherwise an appropriate error code will be returned.

SEE ALSO

`curl_easy_setopt(3)`