

NAME

CURLOPT_TIMEOUT – maximum time the transfer is allowed to complete

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_TIMEOUT, long timeout);
```

DESCRIPTION

Pass a long as parameter containing *timeout* – the maximum time in seconds that you allow the entire transfer operation to take. The whole thing, from start to end. Normally, name lookups can take a considerable time and limiting operations risk aborting perfectly normal operations.

CURLOPT_TIMEOUT_MS(3) is the same function but set in milliseconds.

If both *CURLOPT_TIMEOUT(3)* and *CURLOPT_TIMEOUT_MS(3)* are set, the value set last is used.

Since this option puts a hard limit on how long time a request is allowed to take, it has limited use in dynamic use cases with varying transfer times. That is especially apparent when using the multi interface, which may queue the transfer, and that time is included. You are advised to explore *CURLOPT_LOW_SPEED_LIMIT(3)*, *CURLOPT_LOW_SPEED_TIME(3)* or using *CURLOPT_PROGRESSFUNCTION(3)* to implement your own timeout logic.

The connection timeout set with *CURLOPT_CONNECTTIMEOUT(3)* is included in this general all-covering timeout.

With *CURLOPT_CONNECTTIMEOUT(3)* set to 3 and *CURLOPT_TIMEOUT(3)* set to 5, the operation can never last longer than 5 seconds.

With *CURLOPT_CONNECTTIMEOUT(3)* set to 4 and *CURLOPT_TIMEOUT(3)* set to 2, the operation can never last longer than 2 seconds.

This option may cause libcurl to use the SIGALRM signal to timeout system calls on builds not using async DNS. In Unix-like systems, this might cause signals to be used unless *CURLOPT_NOSIGNAL(3)* is set.

DEFAULT

0 (zero) which means it never times out during transfer.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* complete within 20 seconds */
        curl_easy_setopt(curl, CURLOPT_TIMEOUT, 20L);

        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_CONNECTTIMEOUT(3), CURLOPT_LOW_SPEED_LIMIT(3), CUR-
LOPT_TCP_KEEPALIVE(3), CURLOPT_TIMEOUT_MS(3)