NAME

CURLOPT_TIMEOUT_MS − set maximum time the request is allowed to take

SYNOPSIS

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

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

DESCRIPTION

Pass a long as parameter containing *timeout* - the maximum time in milliseconds that you allow the libcurl transfer operation to take. Normally, name lookups can take a considerable time and limiting operations to less than a few minutes risk aborting perfectly normal operations. This option may cause libcurl to use the SIGALRM signal to timeout system calls.

If libcurl is built to use the standard system name resolver, that portion of the transfer will still use full-second resolution for timeouts with a minimum timeout allowed of one second.

In unix-like systems, this might cause signals to be used unless *CURLOPT_NOSIGNAL(3)* is set.

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

Since this puts a hard limit for how long time a request is allowed to take, it has limited use in dynamic use cases with varying transfer times. You are then advised to explore *CURLOPT_LOW_SPEED_LIMIT(3)*, *CURLOPT_LOW_SPEED_TIME(3)* or using *CURLOPT_PROGRESSFUNCTION(3)* to implement your own timeout logic.

DEFAULT

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

PROTOCOLS

All

EXAMPLE

```
CURL *curl = curl_easy_init();
if(curl) {
  curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");

  /* complete within 20000 milliseconds */
  curl_easy_setopt(curl, CURLOPT_TIMEOUT_MS, 20000L);

  curl_easy_perform(curl);
}
```

AVAILABILITY

Always

RETURN VALUE

Returns CURLE_OK

SEE ALSO

CURLOPT_TIMEOUT(3), **CURLOPT_CONNECTTIMEOUT**(3), **CURLOPT_LOW_SPEED_LIMIT**(3),