

NAME

CURLOPT_TRANSFER_ENCODING – ask for HTTP Transfer Encoding

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_TRANSFER_ENCODING,
                           long enable);
```

DESCRIPTION

Pass a long set to 1L to *enable* or 0 to disable.

Adds a request for compressed Transfer Encoding in the outgoing HTTP request. If the server supports this and so desires, it can respond with the HTTP response sent using a compressed Transfer-Encoding that is automatically uncompressed by libcurl on reception.

Transfer-Encoding differs slightly from the Content-Encoding you ask for with *CURLOPT_ACCEPT_ENCODING(3)* in that a Transfer-Encoding is strictly meant to be for the transfer and thus must be decoded before the data arrives in the client. Traditionally, Transfer-Encoding has been much less used and supported by both HTTP clients and HTTP servers.

DEFAULT

0

PROTOCOLS

This functionality affects http only

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_TRANSFER_ENCODING, 1L);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.21.6

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_ACCEPT_ENCODING(3), CURLOPT_HTTP_TRANSFER_DECODING(3)