

NAME

CURLOPT_FAILONERROR – request failure on HTTP response ≥ 400

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_FAILONERROR, long fail);
```

DESCRIPTION

A long parameter set to 1 tells the library to fail the request if the HTTP code returned is equal to or larger than 400. The default action would be to return the page normally, ignoring that code.

This method is not fail-safe and there are occasions where non-successful response codes slip through, especially when authentication is involved (response codes 401 and 407).

You might get some amounts of headers transferred before this situation is detected, like when a "100-continue" is received as a response to a POST/PUT and a 401 or 407 is received immediately afterwards.

When this option is used and an error is detected, it causes the connection to get closed and `CURLE_HTTP_RETURNED_ERROR` is returned.

DEFAULT

0, do not fail on error

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_FAILONERROR, 1L);
        result = curl_easy_perform(curl);
        if(result == CURLE_HTTP_RETURNED_ERROR) {
            /* an HTTP response error problem */
        }
    }
}
```

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

`CURLINFO_RESPONSE_CODE(3)`, `CURLOPT_HTTP200ALIASES(3)`, `CURLOPT_KEEP_SENDING_ON_ERROR(3)`