

NAME

CURLOPT_POSTREDIR – how to act on an HTTP POST redirect

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_POSTREDIR,
                           long bitmask);
```

DESCRIPTION

Pass a bitmask to control how libcurl acts on redirects after POSTs that get a 301, 302 or 303 response back. A parameter with bit 0 set (value **CURL_REDIR_POST_301**) tells the library to not convert POST requests into GET requests when following a 301 redirection. Setting bit 1 (value **CURL_REDIR_POST_302**) makes libcurl maintain the request method after a 302 redirect whilst setting bit 2 (value **CURL_REDIR_POST_303**) makes libcurl maintain the request method after a 303 redirect. The value **CURL_REDIR_POST_ALL** is a convenience define that sets all three bits.

The non-RFC behavior is ubiquitous in web browsers, so the library does the conversion by default to maintain consistency. A server may require a POST to remain a POST after such a redirection. This option is meaningful only when setting *CURLOPT_FOLLOWLOCATION(3)*.

This option affects transfers where libcurl has been told to use HTTP POST using for example *CURLOPT_POST(3)* or *CURLOPT_MIMEPOST(3)* and not if the method has merely been modified with *CURLOPT_CUSTOMREQUEST(3)*.

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");

        /* a silly POST example */
        curl_easy_setopt(curl, CURLOPT_POSTFIELDS, "data=true");

        /* example.com is redirected, so we tell libcurl to send POST on 301,
           302 and 303 HTTP response codes */
        curl_easy_setopt(curl, CURLOPT_POSTREDIR, CURL_REDIR_POST_ALL);

        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

HISTORY

This option was known as *CURLOPT_POST301* up to 7.19.0 as it only supported the 301 then. *CURL_REDIR_POST_303* was added in 7.26.0.

CURL_REDIR_* macros became *long* types in 8.16.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.19.1

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_EFFECTIVE_METHOD(3), CURLINFO_REDIRECT_COUNT(3), CURLOPT_FOLLOWLOCATION(3), CURLOPT_MAXREDIRS(3), CURLOPT_POSTFIELDS(3)