

NAME

CURLOPT_POSTFIELDSIZE – size of POST data pointed to

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_POSTFIELDSIZE, long size);
```

DESCRIPTION

If you want to post static data to the server without having libcurl do a `strlen()` to measure the data size, this option must be used. When this option is used you can post fully binary data, which otherwise is likely to fail. If this size is set to `-1`, libcurl uses `strlen()` to get the size or relies on the `CURLOPT_READFUNCTION(3)` (if used) to signal the end of data.

If you post more than 2GB, use `CURLOPT_POSTFIELDSIZE_LARGE(3)`.

DEFAULT

`-1`

PROTOCOLS

This functionality affects http, mqtt and rtsp

EXAMPLE

```
#include <string.h> /* for strlen */

int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        const char *data = "data to send";

        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* size of the POST data */
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE, (long)strlen(data));

        curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data);

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

AVAILABILITY

Added in curl 7.2

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_POSTFIELDS(3)`, `CURLOPT_POSTFIELDSIZE_LARGE(3)`