

NAME

CURLOPT_HTTPPOST – multipart formpost content

SYNOPSIS

#include <curl/curl.h>

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HTTPPOST,
                          struct curl_httppost *formpost);
```

DESCRIPTION**This option is deprecated.** Use *CURLOPT_MIMEPOST(3)* instead.

Tells libcurl you want a **multipart/formdata** HTTP POST to be made and you instruct what data to pass on to the server in the *formpost* argument. Pass a pointer to a linked list of *curl_httppost* structs as parameter. The easiest way to create such a list, is to use *curl_formadd(3)* as documented. The data in this list must remain intact as long as the curl transfer is alive and is using it.

Using POST with HTTP 1.1 implies the use of a "Expect: 100–continue" header. You can disable this header with *CURLOPT_HTTPHEADER(3)*.

When setting *CURLOPT_HTTPPOST(3)*, libcurl automatically sets *CURLOPT_NOBODY(3)* to 0.

DEFAULT

NULL

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    struct curl_httppost *formpost;
    struct curl_httppost *lastptr;

    /* Fill in the file upload field. This makes libcurl load data from
       the given filename when curl_easy_perform() is called. */
    curl_formadd(&formpost,
                 &lastptr,
                 CURLFORM_COPYNAME, "sendfile",
                 CURLFORM_FILE, "postit2.c",
                 CURLFORM_END);

    /* Fill in the filename field */
    curl_formadd(&formpost,
                 &lastptr,
                 CURLFORM_COPYNAME, "filename",
                 CURLFORM_COPYCONTENTS, "postit2.c",
                 CURLFORM_END);

    /* Fill in the submit field too, even if this is rarely needed */
    curl_formadd(&formpost,
                 &lastptr,
                 CURLFORM_COPYNAME, "submit",
                 CURLFORM_COPYCONTENTS, "send",
                 CURLFORM_END);

    CURL *curl = curl_easy_init();
```

```
if(curl) {  
    CURLcode result;  
    curl_easy_setopt(curl, CURLOPT_HTTPPOST, formpost);  
    result = curl_easy_perform(curl);  
    curl_easy_cleanup(curl);  
}  
curl_formfree(formpost);  
}
```

DEPRECATED

Deprecated in 7.56.0.

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

CURLOPT_MIMEPOST(3), **CURLOPT_POST(3)**, **CURLOPT_POSTFIELDS(3)**, **curl_formadd(3)**, **curl_formfree(3)**, **curl_mime_init(3)**