NAME

CURLOPT_POSTFIELDS − specify data to POST to server

SYNOPSIS

#include <curl/curl.h>

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_POSTFIELDS, char *postdata);

DESCRIPTION

Pass a char * as parameter, pointing to the full data to send in a HTTP POST operation. You must make sure that the data is formatted the way you want the server to receive it. libcurl will not convert or encode it for you in any way. For example, the web server may assume that this data is url-encoded.

The data pointed to is NOT copied by the library: as a consequence, it must be preserved by the calling application until the associated transfer finishes.

This POST is a normal application/x-www-form-urlencoded kind (and libcurl will set that Content-Type by default when this option is used), which is commonly used by HTML forms. Change Content-Type with *CURLOPT_HTTPHEADER(3)*.

Using *CURLOPT_POSTFIELDS(3)* implies *CURLOPT_POST(3)*.

If you want to do a zero-byte POST, you need to set *CURLOPT_POSTFIELDSIZE(3)* explicitly to zero, as simply setting *CURLOPT_POSTFIELDS(3)* to NULL or "" just effectively disables the sending of the specified string. libcurl will instead assume that you'll send the POST data using the read callback!

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

To make multipart/formdata posts (aka RFC2388-posts), check out the *CURLOPT_HTTPPOST(3)* option combined with *curl_formadd(3)*.

DEFAULT

NULL

PROTOCOLS

HTTP

EXAMPLE

TODO

AVAILABILITY

Always

RETURN VALUE

Returns CURLE_OK

SEE ALSO

CURLOPT_POSTFIELDSIZE(3), **CURLOPT_READFUNCTION**(3),