

NAME

curl_ws_send - send WebSocket data

SYNOPSIS

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

```
CURLcode curl_ws_send(CURL *curl, const void *buffer, size_t buflen,  
                      size_t *sent, curl_off_t fragsize,  
                      unsigned int flags);
```

DESCRIPTION

This function call is EXPERIMENTAL.

Send the specific message fragment over an established WebSocket connection. The *buffer* holds the data to send and it is *buflen* number of payload bytes in that memory area.

sent is returned as the number of payload bytes actually sent.

To send a (huge) fragment using multiple calls with partial content per invoke, set the *CURLWS_OFFSET* bit and the *fragsize* argument as the total expected size for the first part, then set the *CURLWS_OFFSET* with a zero *fragsize* for the following parts.

If not sending a partial fragment or if this is raw mode, *fragsize* should be set to zero.

If **CURLWS_RAW_MODE** is enabled in *CURLOPT_WS_OPTIONS(3)*, the **flags** argument should be set to 0.

To send a message consisting of multiple frames, set the *CURLWS_CONT* bit in all frames except the final one.

FLAGS**CURLWS_TEXT**

The buffer contains text data. Note that this makes a difference to WebSocket but libcurl itself does not make any verification of the content or precautions that you actually send valid UTF-8 content.

CURLWS_BINARY

This is binary data.

CURLWS_CONT

This is not the final fragment of the message, which implies that there is another fragment coming as part of the same message where this bit is not set.

CURLWS_CLOSE

Close this transfer.

CURLWS_PING

This is a ping.

CURLWS_PONG

This is a pong.

CURLWS_OFFSET

The provided data is only a partial fragment and there is more coming in a following call to *curl_ws_send()*. When sending only a piece of the fragment like this, the *fragsize* must be provided with the total expected fragment size in the first call and it needs to be zero in subsequent calls.

EXAMPLE

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

const char *send_payload = "magic";

int main(void)
{
    size_t sent;
    CURLcode res;
    CURL *curl = curl_easy_init();
    curl_easy_setopt(curl, CURLOPT_URL, "wss://example.com/");
    curl_easy_setopt(curl, CURLOPT_CONNECT_ONLY, 2L);
    curl_easy_perform(curl);
    res = curl_ws_send(curl, send_payload, strlen(send_payload), &sent, 0,
                      CURLWS_PING);
    curl_easy_cleanup(curl);
    return (int)res;
}
```

AVAILABILITY

Added in 7.86.0.

RETURN VALUE

CURLE_OK (zero) means that the data was sent properly, non-zero means an error occurred as *<curl/curl.h>* defines. See the *libcurl-errors(3)* man page for the full list with descriptions.

SEE ALSO

curl_easy_setopt(3), curl_easy_perform(3), curl_easy_getinfo(3), curl_ws_recv(3), libcurl-ws(3)