NAME

CURLMOPT_SOCKETFUNCTION − callback informed about what to wait for

SYNOPSIS

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

int socket_callback(CURL *easy,      /* easy handle */
                    curl_socket_t s, /* socket */
                    int what,        /* describes the socket */
                    void *userp,     /* private callback pointer */
                    void *socketp);  /* private socket pointer */

CURLMcode curl_multi_setopt(CURLM *handle, CURLMOPT_SOCKETFUNCTION, socket_callback);
```

DESCRIPTION

Pass a pointer to your callback function, which should match the prototype shown above.

When the *curl_multi_socket_action(3)* function runs, it informs the application about updates in the socket (file descriptor) status by doing none, one, or multiple calls to the **socket_callback**. The callback gets status updates with changes since the previous time the callback was called. If the given callback pointer is NULL, no callback will be called. Set the callback's **userp** argument with *CURLMOPT_SOCKETDATA(3)*. See *curl_multi_socket_action(3)* for more details on how the callback is used and should work.

The **what** parameter informs the callback on the status of the given socket. It can hold one of these values:

CURL_POLL_IN
: Wait for incoming data. For the socket to become readable.

CURL_POLL_OUT
: Wait for outgoing data. For the socket to become writable.

CURL_POLL_INOUT
: Wait for incoming and outgoing data. For the socket to become readable or writable.

CURL_POLL_REMOVE
: The specified socket/file descriptor is no longer used by libcurl.

DEFAULT

NULL (no callback)

PROTOCOLS

All

EXAMPLE

TODO

AVAILABILITY

Added in 7.15.4

RETURN VALUE

Returns CURLM_OK.

SEE ALSO

CURLMOPT_SOCKETDATA(3), **curl_multi_socket_action**(3), **CURLMOPT_TIMERFUNCTION**(3)