

NAME

CURLMOPT_SOCKETDATA – custom pointer passed to the socket callback

SYNOPSIS

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

```
CURLMcode curl_multi_setopt(CURLM *handle, CURLMOPT_SOCKETDATA, void *pointer);
```

DESCRIPTION

A data *pointer* to pass to the socket callback set with the *CURLMOPT_SOCKETFUNCTION(3)* option.

This pointer is not touched by libcurl but is only passed in as the socket callback's **clientp** argument.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
struct priv {
    void *ours;
};

static int sock_cb(CURL *e, curl_socket_t s, int what, void *cbp, void *sockp)
{
    struct priv *p = sockp;
    printf("my ptr: %p\n", p->ours);

    if(what == CURL_POLL_REMOVE) {
        /* remove the socket from our collection */
    }
    if(what & CURL_POLL_IN) {
        /* wait for read on this socket */
    }
    if(what & CURL_POLL_OUT) {
        /* wait for write on this socket */
    }

    return 0;
}

int main(void)
{
    struct priv setup;
    CURLM *multi = curl_multi_init();
    /* ... use socket callback and custom pointer */
    curl_multi_setopt(multi, CURLMOPT_SOCKETFUNCTION, sock_cb);
    curl_multi_setopt(multi, CURLMOPT_SOCKETDATA, &setup);
}
```

AVAILABILITY

Added in curl 7.15.4

RETURN VALUE

Returns CURLM_OK.

SEE ALSO**CURLMOPT_SOCKETFUNCTION(3),
curl_multi_socket_action(3)****CURLMOPT_TIMERFUNCTION(3),**