

NAME

CURLOPT_MAIL_AUTH – SMTP authentication address

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_MAIL_AUTH, char *auth);
```

DESCRIPTION

Pass a pointer to a null-terminated string as parameter. This is used to specify the authentication address (identity) of a submitted message that is being relayed to another server.

This optional parameter allows co-operating agents in a trusted environment to communicate the authentication of individual messages and should only be used by the application program, using libcurl, if the application is itself a mail server acting in such an environment. If the application is operating as such and the AUTH address is not known or is invalid, then an empty string should be used for this parameter.

Unlike *CURLOPT_MAIL_FROM(3)* and *CURLOPT_MAIL_RCPT(3)*, the address should not be specified within a pair of angled brackets (<>). If an empty string is used, a pair of brackets is sent by libcurl as required by RFC 2554.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

DEFAULT

NULL

PROTOCOLS

This functionality affects smtp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "smtp://example.com/");
        curl_easy_setopt(curl, CURLOPT_MAIL_AUTH, "<secret@cave>");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.25.0

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_MAIL_FROM(3), *CURLOPT_MAIL_RCPT(3)*