NAME

CURLOPT_COOKIE − HTTP Cookie header

SYNOPSIS

#include <curl/curl.h>

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_COOKIE, char *cookie);

DESCRIPTION

Pass a pointer to a null−terminated string as parameter. It is used to set one or more cookies in the HTTP request. The format of the string should be NAME=CONTENTS, where NAME is the cookie name and CONTENTS is what the cookie should contain.

To set multiple cookies, set them all using a single option concatenated like this: "name1=content1; name2=content2;" etc. libcurl does not syntax check the data but assumes the application gives it what it needs to send.

This option sets the cookie header explicitly in the outgoing request(s). If multiple requests are done due to authentication, followed redirections or similar, they all get this cookie passed on.

The cookies set by this option are separate from the internal cookie storage held by the cookie engine and they are not be modified by it. If you enable the cookie engine and either you have imported a cookie of the same name (e.g. 'foo') or the server has set one, it has no effect on the cookies you set here. A request to the server sends both the 'foo' held by the cookie engine and the 'foo' held by this option. To set a cookie that is instead held by the cookie engine and can be modified by the server use *CURLOPT_COOKIELIST(3)*.

Since this custom cookie is appended to the Cookie: header in addition to any cookies set by the cookie engine, there is a risk that the header ends up too long and thereby getting the entire request rejected by the server.

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.

This option does not enable the cookie engine. Use *CURLOPT_COOKIEFILE(3)* or *CURLOPT_COOKIEJAR(3)* to enable parsing and sending cookies automatically.

DEFAULT

NULL, no cookies

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode result;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

    curl_easy_setopt(curl, CURLOPT_COOKIE, "tool=curl; fun=yes;");

    result = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
```

```
  }
}
```

AVAILABILITY

Added in curl 7.1

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

CURLINFO_COOKIELIST(3), **CURLOPT_COOKIEFILE**(3), **CURLOPT_COOKIEJAR**(3), **CURLOPT_COOKIELIST**(3), **CURLOPT_HTTPHEADER**(3)