

NAME

curl_easy_getinfo - extract information from a curl handle

SYNOPSIS

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

```
CURLcode curl_easy_getinfo(CURL *curl, CURLINFO info, ... );
```

DESCRIPTION

Get the *info* kept in the *curl* handle. The third argument **MUST** be pointing to the specific type of the used option which is documented in each man page of the *info* option. The data is stored accordingly and can be relied upon only if this function returns CURLE_OK. Use this function after a performed transfer if you want to get transfer related data.

You should not free the memory returned by this function unless it is explicitly mentioned below.

OPTIONS

The following information can be extracted:

CURLINFO_ACTIVESOCKET

The session's active socket. See *CURLINFO_ACTIVESOCKET(3)*

CURLINFO_APPCONNECT_TIME

The time it took from the start until the SSL connect/handshake with the remote host was completed as a double in number of seconds. (Added in 7.19.0)

CURLINFO_APPCONNECT_TIME_T

The time it took from the start until the SSL connect/handshake with the remote host was completed in number of microseconds. (Added in 7.60.0) See *CURLINFO_APPCONNECT_TIME_T(3)*

CURLINFO_CAINFO

Get the default value for *CURLOPT_CAINFO(3)*. See *CURLINFO_CAINFO(3)*

CURLINFO_CAPATH

Get the default value for *CURLOPT_CAPATH(3)*. See *CURLINFO_CAPATH(3)*

CURLINFO_CERTINFO

Certificate chain. See *CURLINFO_CERTINFO(3)*

CURLINFO_CONDITION_UNMET

Whether or not a time conditional was met or 304 HTTP response. See *CURLINFO_CONDITION_UNMET(3)*

CURLINFO_CONNECT_TIME

The time it took from the start until the connect to the remote host (or proxy) was completed. As a double. See *CURLINFO_CONNECT_TIME(3)*

CURLINFO_CONNECT_TIME_T

The time it took from the start until the connect to the remote host (or proxy) was completed. In microseconds. See *CURLINFO_CONNECT_TIME_T(3)*.

CURLINFO_CONN_ID

The ID of the last connection used by the transfer. (Added in 8.2.0) See *CURLINFO_CONN_ID(3)*

CURLINFO_CONTENT_LENGTH_DOWNLOAD

(Deprecated) Content length from the Content-Length header. See *CURLINFO_CONTENT_LENGTH_DOWNLOAD(3)*

CURLINFO_CONTENT_LENGTH_DOWNLOAD_T

Content length from the Content-Length header. See *CURLINFO_CONTENT_LENGTH_DOWNLOAD_T(3)*

CURLINFO_CONTENT_LENGTH_UPLOAD

(Deprecated) Upload size. See *CURLINFO_CONTENT_LENGTH_UPLOAD(3)*

CURLINFO_CONTENT_LENGTH_UPLOAD_T

Upload size. See *CURLINFO_CONTENT_LENGTH_UPLOAD_T(3)*

CURLINFO_CONTENT_TYPE

Content type from the *Content-Type:* header. We recommend using *curl_easy_header(3)* instead. See *CURLINFO_CONTENT_TYPE(3)*

CURLINFO_COOKIELIST

List of all known cookies. See *CURLINFO_COOKIELIST(3)*

CURLINFO_EFFECTIVE_METHOD

Last used HTTP method. See *CURLINFO_EFFECTIVE_METHOD(3)*

CURLINFO_EFFECTIVE_URL

Last used URL. See *CURLINFO_EFFECTIVE_URL(3)*

CURLINFO_FILETIME

Remote time of the retrieved document. See *CURLINFO_FILETIME(3)*

CURLINFO_FILETIME_T

Remote time of the retrieved document. See *CURLINFO_FILETIME_T(3)*

CURLINFO_FTP_ENTRY_PATH

The entry path after logging in to an FTP server. See *CURLINFO_FTP_ENTRY_PATH(3)*

CURLINFO_HEADER_SIZE

Number of bytes of all headers received. See *CURLINFO_HEADER_SIZE(3)*

CURLINFO_HTTPAUTH_AVAIL

Available HTTP authentication methods. See *CURLINFO_HTTPAUTH_AVAIL(3)*

CURLINFO_HTTP_CONNECTCODE

Last proxy CONNECT response code. See *CURLINFO_HTTP_CONNECTCODE(3)*

CURLINFO_HTTP_VERSION

The http version used in the connection. See *CURLINFO_HTTP_VERSION(3)*

CURLINFO_LASTSOCKET

(Deprecated) Last socket used. See *CURLINFO_LASTSOCKET(3)*

CURLINFO_LOCAL_IP

Source IP address of the last connection. See *CURLINFO_LOCAL_IP(3)*

CURLINFO_LOCAL_PORT

Source port number of the last connection. See *CURLINFO_LOCAL_PORT(3)*

CURLINFO_NAMELOOKUP_TIME

Time from start until name resolving completed as a double. See *CURLINFO_NAMELOOKUP_TIME(3)*

CURLINFO_NAMELOOKUP_TIME_T

Time from start until name resolving completed in number of microseconds. See *CURLINFO_NAMELOOKUP_TIME_T(3)*

CURLINFO_NUM_CONNECTS

Number of new successful connections used for previous transfer. See *CURLINFO_NUM_CONNECTS(3)*

CURLINFO_OS_ERRNO

The errno from the last failure to connect. See *CURLINFO_OS_ERRNO(3)*

CURLINFO_POSTTRANSFER_TIME_T

The time it took from the start until the last byte is sent by libcurl. In microseconds. (Added in 8.10.0) See *CURLINFO_POSTTRANSFER_TIME_T(3)*

CURLINFO_PRETRANSFER_TIME

The time it took from the start until the file transfer is just about to begin. This includes all pre-transfer commands and negotiations that are specific to the particular protocol(s) involved. See *CURLINFO_PRETRANSFER_TIME(3)*

CURLINFO_PRETRANSFER_TIME_T

The time it took from the start until the file transfer is just about to begin. This includes all pre-transfer commands and negotiations that are specific to the particular protocol(s) involved. In microseconds. See *CURLINFO_PRETRANSFER_TIME_T(3)*

CURLINFO_PRIMARY_IP

Destination IP address of the last connection. See *CURLINFO_PRIMARY_IP(3)*

CURLINFO_PRIMARY_PORT

Destination port of the last connection. See *CURLINFO_PRIMARY_PORT(3)*

CURLINFO_PRIVATE

User's private data pointer. See *CURLINFO_PRIVATE(3)*

CURLINFO_PROTOCOL

(Deprecated) The protocol used for the connection. (Added in 7.52.0) See *CURLINFO_PROTOCOL(3)*

CURLINFO_PROXYAUTH_AVAIL

Available HTTP proxy authentication methods. See *CURLINFO_PROXYAUTH_AVAIL(3)*

CURLINFO_PROXY_ERROR

Detailed proxy error. See *CURLINFO_PROXY_ERROR(3)*

CURLINFO_PROXY_SSL_VERIFYRESULT

Proxy certificate verification result. See *CURLINFO_PROXY_SSL_VERIFYRESULT(3)*

CURLINFO_QUEUE_TIME_T

The time during which the transfer was held in a waiting queue before it could start for real in number of microseconds. (Added in 8.6.0) See *CURLINFO_QUEUE_TIME_T(3)*

CURLINFO_REDIRECT_COUNT

Total number of redirects that were followed. See *CURLINFO_REDIRECT_COUNT(3)*

CURLINFO_REDIRECT_TIME

The time it took for all redirection steps include name lookup, connect, pretransfer and transfer before final transaction was started. So, this is zero if no redirection took place. As a double. See *CURLINFO_REDIRECT_TIME(3)*

CURLINFO_REDIRECT_TIME_T

The time it took for all redirection steps include name lookup, connect, pretransfer and transfer before final transaction was started. So, this is zero if no redirection took place. In number of microseconds. See *CURLINFO_REDIRECT_TIME_T(3)*

CURLINFO_REDIRECT_URL

URL a redirect would take you to, had you enabled redirects. See *CURLINFO_REDIRECT_URL(3)*

CURLINFO_REFERER

Referrer header. See *CURLINFO_REFERER(3)*

CURLINFO_REQUEST_SIZE

Number of bytes sent in the issued HTTP requests. See *CURLINFO_REQUEST_SIZE(3)*

CURLINFO_RESPONSE_CODE

Last received response code. See *CURLINFO_RESPONSE_CODE(3)*

CURLINFO_RETRY_AFTER

The value from the Retry-After header. See *CURLINFO_RETRY_AFTER(3)*

CURLINFO_RTSP_CLIENT_CSEQ

The RTSP client CSeq that is expected next. See *CURLINFO_RTSP_CLIENT_CSEQ(3)*

CURLINFO_RTSP_CSEQ_RECV

RTSP CSeq last received. See *CURLINFO_RTSP_CSEQ_RECV(3)*

CURLINFO_RTSP_SERVER_CSEQ

The RTSP server CSeq that is expected next. See *CURLINFO_RTSP_SERVER_CSEQ(3)*

CURLINFO_RTSP_SESSION_ID

RTSP session ID. See *CURLINFO_RTSP_SESSION_ID(3)*

CURLINFO_SCHEME

The scheme used for the connection. (Added in 7.52.0) See *CURLINFO_SCHEME(3)*

CURLINFO_SIZE_DOWNLOAD

(Deprecated) Number of bytes downloaded. See *CURLINFO_SIZE_DOWNLOAD(3)*

CURLINFO_SIZE_DOWNLOAD_T

Number of bytes downloaded. See *CURLINFO_SIZE_DOWNLOAD_T(3)*

CURLINFO_SIZE_UPLOAD

(Deprecated) Number of bytes uploaded. See *CURLINFO_SIZE_UPLOAD(3)*

CURLINFO_SIZE_UPLOAD_T

Number of bytes uploaded. See *CURLINFO_SIZE_UPLOAD_T(3)*

CURLINFO_SPEED_DOWNLOAD

(Deprecated) Average download speed. See *CURLINFO_SPEED_DOWNLOAD(3)*

CURLINFO_SPEED_DOWNLOAD_T

Average download speed. See *CURLINFO_SPEED_DOWNLOAD_T(3)*

CURLINFO_SPEED_UPLOAD

(Deprecated) Average upload speed. See *CURLINFO_SPEED_UPLOAD(3)*

CURLINFO_SPEED_UPLOAD_T

Average upload speed in number of bytes per second. See *CURLINFO_SPEED_UPLOAD_T(3)*

CURLINFO_SSL_ENGINES

A list of OpenSSL crypto engines. See *CURLINFO_SSL_ENGINES(3)*

CURLINFO_SSL_VERIFYRESULT

Certificate verification result. See *CURLINFO_SSL_VERIFYRESULT(3)*

CURLINFO_STARTTRANSFER_TIME

The time it took from the start until the first byte is received by libcurl. As a double. See *CURLINFO_STARTTRANSFER_TIME(3)*

CURLINFO_STARTTRANSFER_TIME_T

The time it took from the start until the first byte is received by libcurl. In microseconds. See *CURLINFO_STARTTRANSFER_TIME_T(3)*

CURLINFO_TLS_SESSION

(Deprecated) TLS session info that can be used for further processing. See *CURLINFO_TLS_SESSION(3)*. Use *CURLINFO_TLS_SSL_PTR(3)* instead!

CURLINFO_TLS_SSL_PTR

TLS session info that can be used for further processing. See *CURLINFO_TLS_SSL_PTR(3)*

CURLINFO_TOTAL_TIME

Total time of previous transfer. See *CURLINFO_TOTAL_TIME(3)*

CURLINFO_TOTAL_TIME_T

Total time of previous transfer. See *CURLINFO_TOTAL_TIME_T(3)*

CURLINFO_USED_PROXY

Whether the proxy was used (Added in 8.7.0). See *CURLINFO_USED_PROXY(3)*

CURLINFO_XFER_ID

The ID of the transfer. (Added in 8.2.0) See *CURLINFO_XFER_ID(3)*

TIMES

An overview of the time values available from *curl_easy_getinfo(3)*

curl_easy_perform()

```
|
|--QUEUE
|--|--NAMELOOKUP
|--|--|--CONNECT
|--|--|--APPCONNECT
|--|--|--|--PRETRANSFER
|--|--|--|--POSTTRANSFER
|--|--|--|--|--STARTTRANSFER
|--|--|--|--|--TOTAL
```

|--|--|--|--|--|REDIRECT

*CURLINFO_QUEUE_TIME_T(3), CURLINFO_NAMELOOKUP_TIME_T(3),
CURLINFO_CONNECT_TIME_T(3), CURLINFO_APPCONNECT_TIME_T(3),
CURLINFO_PRETRANSFER_TIME_T(3), CURLINFO_POSTTRANSFER_TIME_T(3),
CURLINFO_STARTTRANSFER_TIME_T(3), CURLINFO_TOTAL_TIME_T(3),
CURLINFO_REDIRECT_TIME_T(3)*

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

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

        if(CURLE_OK == res) {
            char *ct;
            /* ask for the content-type */
            res = curl_easy_getinfo(curl, CURLINFO_CONTENT_TYPE, &ct);

            if((CURLE_OK == res) && ct)
                printf("We received Content-Type: %s\n", ct);
        }

        /* always cleanup */
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.4.1

RETURN VALUE

If the operation was successful, CURLE_OK is returned. Otherwise an appropriate error code is returned.

SEE ALSO**curl_easy_setopt(3)**