

NAME

CURLOPT_RTSP_REQUEST – RTSP request

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_RTSP_REQUEST, long request);
```

DESCRIPTION

Tell libcurl what kind of RTSP request to make. Pass one of the following RTSP enum values as a long in the *request* argument. Unless noted otherwise, commands require the Session ID to be initialized.

CURL_RTSPREQ_OPTIONS

Used to retrieve the available methods of the server. The application is responsible for parsing and obeying the response. The session ID is not needed for this method.

CURL_RTSPREQ_DESCRIBE

Used to get the low level description of a stream. The application should note what formats it understands in the 'Accept:' header. Unless set manually, libcurl automatically adds in 'Accept: *application/sdp*'. Time-condition headers are added to Describe requests if the *CURLOPT_TIMECONDITION(3)* option is used. (The session ID is not needed for this method)

CURL_RTSPREQ_ANNOUNCE

When sent by a client, this method changes the description of the session. For example, if a client is using the server to record a meeting, the client can use Announce to inform the server of all the meta-information about the session. ANNOUNCE acts like an HTTP PUT or POST like *CURL_RTSPREQ_SET_PARAMETER*

CURL_RTSPREQ_SETUP

Setup is used to initialize the transport layer for the session. The application must set the desired Transport options for a session by using the *CURLOPT_RTSP_TRANSPORT(3)* option prior to calling setup. If no session ID is currently set with *CURLOPT_RTSP_SESSION_ID(3)*, libcurl extracts and uses the session ID in the response to this request. The session ID is not needed for this method.

CURL_RTSPREQ_PLAY

Send a Play command to the server. Use the *CURLOPT_RANGE(3)* option to modify the playback time (e.g. *npt=10-15*).

CURL_RTSPREQ_PAUSE

Send a Pause command to the server. Use the *CURLOPT_RANGE(3)* option with a single value to indicate when the stream should be halted. (e.g. *npt=25*)

CURL_RTSPREQ_TEARDOWN

This command terminates an RTSP session. Closing a connection does not terminate the RTSP session since it is valid to control an RTSP session over different connections.

CURL_RTSPREQ_GET_PARAMETER

Retrieve a parameter from the server. By default, libcurl adds a *Content-Type: text/parameters* header on all non-empty requests unless a custom one is set. GET_PARAMETER acts like an HTTP PUT or POST (see *CURL_RTSPREQ_SET_PARAMETER*). Applications wishing to send a heartbeat message (e.g. in the presence of a server-specified timeout) should send use an empty GET_PARAMETER request.

CURL_RTSPREQ_SET_PARAMETER

Set a parameter on the server. By default, libcurl uses a **Content-Type: text/parameters** header unless a custom one is set. The interaction with SET_PARAMETER is much like an HTTP PUT or POST. An application may either use *CURLOPT_UPLOAD(3)* with *CURLOPT_READDATA(3)* like an HTTP PUT, or it may use *CURLOPT_POSTFIELDS(3)* like an HTTP POST. No chunked transfers are allowed, so the application must set the *CURLOPT_INFILESIZE(3)* in the former and *CURLOPT_POSTFIELDSIZE(3)* in the latter. Also, there is no use of multi-part POSTs within

RTSP.

CURL_RTSPREQ_RECORD

Used to tell the server to record a session. Use the *CURLOPT_RANGE(3)* option to modify the record time.

CURL_RTSPREQ_RECEIVE

This is a special request because it does not send any data to the server. The application may call this function in order to receive interleaved RTP data. It returns after processing one read buffer of data in order to give the application a chance to run.

DEFAULT PROTOCOLS

This functionality affects rtsp only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "rtsp://example.com/");
        /* ask for options */
        curl_easy_setopt(curl, CURLOPT_RTSP_REQUEST, CURL_RTSPREQ_OPTIONS);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

HISTORY

CURL_RTSPREQ_* enums became *long* types in 8.13.0, prior to this version a *long* cast was necessary when passed to *curl_easy_setopt(3)*.

AVAILABILITY

Added in curl 7.20.0

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_RTSP_SESSION_ID(3), **CURLOPT_RTSP_STREAM_URI(3)**