

NAME

CURLOPT_PROXY_SSLCERT – HTTPS proxy client certificate

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_PROXY_SSLCERT, char *cert);
```

DESCRIPTION

This option is for connecting to an HTTPS proxy, not an HTTPS server.

Pass a pointer to a null-terminated string as parameter. The string should be the filename of your client certificate used to connect to the HTTPS proxy. The default format "PEM", and can be changed with *CURLOPT_PROXY_SSLCERTTYPE(3)*.

When using a client certificate, you most likely also need to provide a private key with *CURLOPT_PROXY_SSLKEY(3)*.

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 all TLS based protocols: HTTPS, FTPS, IMAPS, POP3S, SMTPS etc.

This option works only with the following TLS backends: GnuTLS, OpenSSL, Schannel, mbedTLS and wolfSSL

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_PROXY, "https://proxy.example");
        curl_easy_setopt(curl, CURLOPT_PROXY_SSLCERT, "client.pem");
        curl_easy_setopt(curl, CURLOPT_PROXY_SSLKEY, "key.pem");
        curl_easy_setopt(curl, CURLOPT_PROXY_KEYPASSWD, "s3cret");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.52.0

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_PROXY_SSLCERTTYPE(3), CURLOPT_PROXY_SSLKEY(3), CURLOPT_PROXY_SSLKEYTYPE(3)