

NAME

CURLINFO_TLS_SESSION – TLS session info

SYNOPSIS

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

```
CURLcode curl_easy_getinfo(CURL *handle, CURLINFO_TLS_SESSION,
                           struct curl_tlssessioninfo **session);
```

DESCRIPTION

This option has been superseded by *CURLINFO_TLS_SSL_PTR(3)*.

This option is exactly the same as *CURLINFO_TLS_SSL_PTR(3)* except in the case of OpenSSL and wolfSSL. If the session *backend* is *CURLSSLBACKEND_OPENSSL* the session *internals* pointer varies depending on the option:

OpenSSL

CURLINFO_TLS_SESSION(3) OpenSSL session *internals* is **SSL_CTX** *.

CURLINFO_TLS_SSL_PTR(3) OpenSSL session *internals* is **SSL** *.

You can obtain an **SSL_CTX** pointer from an SSL pointer using OpenSSL function *SSL_get_SSL_CTX(3)*. Therefore unless you need compatibility with older versions of libcurl use *CURLINFO_TLS_SSL_PTR(3)*. Refer to that document for more information.

wolfSSL

CURLINFO_TLS_SESSION(3) wolfSSL session *internals* is **WOLFSSL_CTX** *.

CURLINFO_TLS_SSL_PTR(3) wolfSSL session *internals* is **WOLFSSL** *.

You can obtain an **WOLFSSL_CTX** pointer from an SSL pointer using wolfSSL function *wolfSSL_get_SSL_CTX(3)*. Therefore unless you need compatibility with older versions of libcurl use *CURLINFO_TLS_SSL_PTR(3)*. Refer to that document for more information.

PROTOCOLS

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

This option works only with the following TLS backends: GnuTLS and OpenSSL

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        struct curl_tlssessioninfo *tls;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
        result = curl_easy_perform(curl);
        if(result != CURLE_OK)
            printf("error: %s\n", curl_easy_strerror(result));
        curl_easy_getinfo(curl, CURLINFO_TLS_SESSION, &tls);
        curl_easy_cleanup(curl);
    }
}
```

DEPRECATED

Deprecated since 7.48.0

AVAILABILITY

Added in curl 7.34.0

RETURN VALUE

curl_easy_getinfo(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_TLS_SSL_PTR(3), **curl_easy_getinfo(3)**, **curl_easy_setopt(3)**