

**NAME**

CURLINFO\_TLS\_SESSION – get TLS session info

**SYNOPSIS**

#include &lt;curl/curl.h&gt;

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

**DESCRIPTION**

Pass a pointer to a 'struct curl\_tlssessioninfo \*'. The pointer will be initialized to refer to a 'struct curl\_tlssessioninfo \*' that will contain an enum indicating the SSL library used for the handshake and the respective internal TLS session structure of this underlying SSL library.

This may then be used to extract certificate information in a format convenient for further processing, such as manual validation. NOTE: this option may not be available for all SSL backends; unsupported SSL backends will always return NULL in the *internals* pointer to indicate that they are not supported.

```
struct curl_tlssessioninfo {
    curl_sslbackend backend;
    void *internals;
};
```

The *backend* struct member is one of the defines in the CURLSSLBACKEND\_\* series: CURLSSLBACKEND\_NONE (when built without TLS support), CURLSSLBACKEND\_OPENSSL, CURLSSLBACKEND\_GNUTLS, CURLSSLBACKEND\_NSS, CURLSSLBACKEND\_GSKIT, CURLSSLBACKEND\_POLARSSL, CURLSSLBACKEND\_CYASSL, CURLSSLBACKEND\_SCHANNEL, CURLSSLBACKEND\_DARWINSSL or CURLSSLBACKEND\_AXTLS. (Note that the OpenSSL forks are all reported as just OpenSSL here.)

The *internals* struct member will point to a TLS library specific pointer with the following underlying types:

OpenSSL	SSL_CTX *
GnuTLS	gnutls_session_t
NSS	PRFileDesc *
gskit	gsk_handle

**PROTOCOLS**

All TLS-based

**EXAMPLE**

TODO

**AVAILABILITY**

Added in 7.34.0

**RETURN VALUE**

Returns CURLE\_OK if the option is supported, and CURLE\_UNKNOWN\_OPTION if not.

**SEE ALSO**

curl\_easy\_getinfo(3), curl\_easy\_setopt(3),