NAME

    SSL_get_session, SSL_get0_session, SSL_get1_session - retrieve TLS/SSL session data

SYNOPSIS

    #include <openssl/ssl.h>

    SSL_SESSION *SSL_get_session(const SSL *ssl);
    SSL_SESSION *SSL_get0_session(const SSL *ssl);
    SSL_SESSION *SSL_get1_session(SSL *ssl);

DESCRIPTION

    SSL_get_session() returns a pointer to the **SSL_SESSION** actually used in **ssl**. The reference count of the **SSL_SESSION** is not incremented, so that the pointer can become invalid by other operations.

    SSL_get0_session() is the same as **SSL_get_session()**.

    SSL_get1_session() is the same as **SSL_get_session()**, but the reference count of the **SSL_SESSION** is incremented by one.

NOTES

    The ssl session contains all information required to re-establish the connection without a full handshake for SSL versions up to and including TLSv1.2. In TLSv1.3 the same is true, but sessions are established after the main handshake has occurred. The server will send the session information to the client at a time of its choosing, which may be some while after the initial connection is established (or never). Calling these functions on the client side in TLSv1.3 before the session has been established will still return an SSL_SESSION object but that object cannot be used for resuming the session. See **SSL_SESSION_is_resumable**(3) for information on how to determine whether an SSL_SESSION object can be used for resumption or not.

    Additionally, in TLSv1.3, a server can send multiple messages that establish a session for a single connection. In that case, on the client side, the above functions will only return information on the last session that was received. On the server side they will only return information on the last session that was sent, or if no session tickets were sent then the session for the current connection.

    The preferred way for applications to obtain a resumable SSL_SESSION object is to use a new session callback as described in **SSL_CTX_sess_set_new_cb**(3). The new session callback is only invoked when a session is actually established, so this avoids the problem described above where an application obtains an SSL_SESSION object that cannot be used for resumption in TLSv1.3. It also enables applications to obtain information about all sessions sent by the server.

A session will be automatically removed from the session cache and marked as non-resumable if the connection is not closed down cleanly, e.g. if a fatal error occurs on the connection or **SSL_shutdown**(3) is not called prior to **SSL_free**(3).

In TLSv1.3 it is recommended that each SSL_SESSION object is only used for resumption once.

SSL_get0_session() returns a pointer to the actual session. As the reference counter is not incremented, the pointer is only valid while the connection is in use. If **SSL_clear**(3) or **SSL_free**(3) is called, the session may be removed completely (if considered bad), and the pointer obtained will become invalid. Even if the session is valid, it can be removed at any time due to timeout during **SSL_CTX_flush_sessions**(3).

If the data is to be kept, **SSL_get1_session()** will increment the reference count, so that the session will not be implicitly removed by other operations but stays in memory. In order to remove the session **SSL_SESSION_free**(3) must be explicitly called once to decrement the reference count again.

SSL_SESSION objects keep internal link information about the session cache list, when being inserted into one SSL_CTX object's session cache. One SSL_SESSION object, regardless of its reference count, must therefore only be used with one SSL_CTX object (and the SSL objects created from this SSL_CTX object).

RETURN VALUES

The following return values can occur:

NULL

There is no session available in **ssl**.

Pointer to an SSL_SESSION

The return value points to the data of an SSL session.

SEE ALSO

ssl(7), **SSL_free**(3), **SSL_clear**(3), **SSL_SESSION_free**(3)

COPYRIGHT

Copyright 2000-2021 The OpenSSL Project Authors. All Rights Reserved.