

**NAME**

SSL\_CTX\_set\_session\_id\_context, SSL\_set\_session\_id\_context - set context within which session can be reused (server side only)

**SYNOPSIS**

```
#include <openssl/ssl.h>
```

```
int SSL_CTX_set_session_id_context(SSL_CTX *ctx, const unsigned char *sid_ctx,  
                                  unsigned int sid_ctx_len);  
int SSL_set_session_id_context(SSL *ssl, const unsigned char *sid_ctx,  
                              unsigned int sid_ctx_len);
```

**DESCRIPTION**

**SSL\_CTX\_set\_session\_id\_context()** sets the context **sid\_ctx** of length **sid\_ctx\_len** within which a session can be reused for the **ctx** object.

**SSL\_set\_session\_id\_context()** sets the context **sid\_ctx** of length **sid\_ctx\_len** within which a session can be reused for the **ssl** object.

**NOTES**

Sessions are generated within a certain context. When exporting/importing sessions with **i2d\_SSL\_SESSION/d2i\_SSL\_SESSION** it would be possible, to re-import a session generated from another context (e.g. another application), which might lead to malfunctions. Therefore, each application must set its own session id context **sid\_ctx** which is used to distinguish the contexts and is stored in exported sessions. The **sid\_ctx** can be any kind of binary data with a given length, it is therefore possible to use e.g. the name of the application and/or the hostname and/or service name ...

The session id context becomes part of the session. The session id context is set by the SSL/TLS server. The **SSL\_CTX\_set\_session\_id\_context()** and **SSL\_set\_session\_id\_context()** functions are therefore only useful on the server side.

OpenSSL clients will check the session id context returned by the server when reusing a session.

The maximum length of the **sid\_ctx** is limited to **SSL\_MAX\_SID\_CTX\_LENGTH**.

**WARNINGS**

If the session id context is not set on an SSL/TLS server and client certificates are used, stored sessions will not be reused but a fatal error will be flagged and the handshake will fail.

If a server returns a different session id context to an OpenSSL client when reusing a session, an error

will be flagged and the handshake will fail. OpenSSL servers will always return the correct session id context, as an OpenSSL server checks the session id context itself before reusing a session as described above.

## RETURN VALUES

**SSL\_CTX\_set\_session\_id\_context()** and **SSL\_set\_session\_id\_context()** return the following values:

- 0 The length **sid\_ctx\_len** of the session id context **sid\_ctx** exceeded the maximum allowed length of **SSL\_MAX\_SID\_CTX\_LENGTH**. The error is logged to the error stack.
- 1 The operation succeeded.

## SEE ALSO

**ssl(7)**

## COPYRIGHT

Copyright 2001-2020 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.