NAME

ne_session_create, ne_close_connection, ne_session_destroy - set up HTTP sessions

SYNOPSIS

#include <ne_session.h>

ne_session *ne_session_create(const char *_scheme_, const char *_hostname_, unsigned int _port_);

void ne_close_connection(ne_session *_session_);

void ne_session_destroy(ne_session *_session_);

DESCRIPTION

An **ne_session** object represents an HTTP session - a logical grouping of a sequence of HTTP requests made to a certain server. Any requests made using the session can use a persistent connection, share cached authentication credentials and any other common attributes.

A new HTTP session is created using the **ne_session_create** function; the *hostname* and *port* parameters specify the origin server to use, along with the *scheme* (usually "http"). Before the first use of **ne_session_create** in a process, ne_sock_init must have been called to perform any global initialization needed by any libraries used by neon.

To enable SSL/TLS for the session, pass the string "https" as the *scheme* parameter, and either register a certificate verification function (see ne_ssl_set_verify) or trust the appropriate certificate (see ne_ssl_trust_cert, ne_ssl_trust_default_ca).

To use a proxy server for the session, it must be configured (see ne_session_proxy) before any requests are created from session object.

Further per-session options may be changed using the ne_set_session_flag interface.

If it is known that the session will not be used for a significant period of time, **ne_close_connection** can be called to close the connection, if one remains open. Use of this function is entirely optional, but it must not be called if there is a request active using the session.

Once a session has been completed, **ne_session_destroy** must be called to destroy the resources associated with the session. Any subsequent use of the session pointer produces undefined behaviour. The session object must not be destroyed until after all associated request objects have been destroyed.

NOTES

The hostname passed to **ne_session_create** is resolved when the first request using the session is dispatched; a DNS resolution failure can only be detected at that time (using the NE_LOOKUP error code); see ne_request_dispatch for details.

RETURN VALUES

ne_session_create will return a pointer to a new session object (and never NULL).

EXAMPLES

Create and destroy a session:

```
ne_session *sess;
sess = ne_session_create("http", "host.example.com", 80);
/* ... use sess ... */
ne_session_destroy(sess);
```

SEE ALSO

ne_ssl_set_verify, ne_ssl_trust_cert, ne_sock_init, ne_set_session_flag

AUTHOR

Joe Orton

Author.

COPYRIGHT