

NAME

CURLOPT_SSL_SESSIONID_CACHE – use the SSL session-ID cache

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSL_SESSIONID_CACHE,
                           long enabled);
```

DESCRIPTION

Pass a long set to 0 to disable libcurl's use of SSL session-ID caching. Set this to 1 to enable it. By default all transfers are done using the cache enabled. While nothing ever should get hurt by attempting to reuse SSL session-IDs, there seem to be or have been broken SSL implementations in the wild that may require you to disable this in order for you to succeed.

DEFAULT

1

PROTOCOLS

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

All TLS backends support this option.

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();
    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/");
        /* switch off session-id use */
        curl_easy_setopt(curl, CURLOPT_SSL_SESSIONID_CACHE, 0L);
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.16.0

RETURN VALUE

curl_easy_setopt(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

**CURLOPT_DNS_CACHE_TIMEOUT(3), CURLOPT_MAXAGE_CONN(3), CUR-
LOPT_MAXLIFETIME_CONN(3), CURLOPT_SSLVERSION(3)**