NAME

CURLOPT_CA_CACHE_TIMEOUT − life−time for cached certificate stores

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_CA_CACHE_TIMEOUT, long age);
```

DESCRIPTION

Pass a long, this sets the timeout in seconds. This tells libcurl the maximum time any cached CA certificate store it has in memory may be kept and reused for new connections. Once the timeout has expired, a subsequent fetch requiring a CA certificate has to reload it.

Building a CA certificate store from a *CURLOPT_CAINFO(3)* file is a slow operation so curl may cache the generated certificate store internally to speed up future connections.

Set the timeout to zero to completely disable caching, or set to −1 to retain the cached store remain forever. By default, libcurl caches this info for 24 hours.

DEFAULT

86400 (24 hours)

PROTOCOLS

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

This option works only with the following TLS backends: GnuTLS, OpenSSL, Schannel and wolfSSL

EXAMPLE

```
int main(void)
{
 CURL *curl = curl_easy_init();
 if(curl) {
  CURLcode result;
  curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/foo.bin");

  /* only reuse certificate stores for a short time */
  curl_easy_setopt(curl, CURLOPT_CA_CACHE_TIMEOUT, 60L);

  result = curl_easy_perform(curl);

  /* in this second request, the cache is not used if more than
     sixty seconds passed since the previous connection */
  result = curl_easy_perform(curl);

  curl_easy_cleanup(curl);
 }
}
```

HISTORY

This option is supported by OpenSSL and its forks (since 7.87.0), Schannel (since 8.5.0), wolfSSL (since 8.9.0) and GnuTLS (since 8.9.0).

AVAILABILITY

Added in curl 7.87.0

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_CAINFO(3), **CURLOPT_CAINFO_BLOB**(3), **CURLOPT_CAPATH**(3), **CURLOPT_SSL_VERIFYHOST**(3), **CURLOPT_SSL_VERIFYPEER**(3)