NAME

CURLOPT_SSLVERSION − set preferred TLS/SSL version

SYNOPSIS

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

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSLVERSION, long version);
```

DESCRIPTION

Pass a long as parameter to control which version of SSL/TLS to attempt to use.

Use one of the available defines for this purpose. The available options are:

CURL_SSLVERSION_DEFAULT
: The default action. This will attempt to figure out the remote SSL protocol version.

CURL_SSLVERSION_TLSv1
: TLSv1.x

CURL_SSLVERSION_SSLv2
: SSLv2

CURL_SSLVERSION_SSLv3
: SSLv3

CURL_SSLVERSION_TLSv1_0
: TLSv1.0 (Added in 7.34.0)

CURL_SSLVERSION_TLSv1_1
: TLSv1.1 (Added in 7.34.0)

CURL_SSLVERSION_TLSv1_2
: TLSv1.2 (Added in 7.34.0)

DEFAULT

CURL_SSLVERSION_DEFAULT

PROTOCOLS

All TLS based protocols: HTTPS, FTPS, IMAPS, POP3, SMTPS etc.

EXAMPLE

```
CURL *curl = curl_easy_init();
if(curl) {
 curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

 /* ask libcurl to use TLS version 1.0 or later */
 curl_easy_setopt(curl, CURLOPT_SSLVERSION, CURL_SSLVERSION_TLSv1);

 /* Perform the request */
 curl_easy_perform(curl);
}
```

AVAILABILITY

SSLv2 is disabled by default since 7.18.1. Other SSL versions availability may vary depending on which backend libcurl has been built to use.

SSLv3 is disabled by default since 7.39.0.

RETURN VALUE

Returns CURLE_OK if the option is supported, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO

CURLOPT_USE_SSL(3), **CURLOPT_HTTP_VERSION**(3), **CURLOPT_IPRESOLVE**(3)

