

NAME

CURLOPT_DOH_SSL_VERIFYHOST – verify the hostname in the DoH SSL certificate

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_DOH_SSL_VERIFYHOST,
                           long verify);
```

DESCRIPTION

Pass a long set to 2L as asking curl to *verify* the DoH (DNS-over-HTTPS) server's certificate name fields against the hostname.

This option is the DoH equivalent of *CURLOPT_SSL_VERIFYHOST(3)* and only affects requests to the DoH server.

When *CURLOPT_DOH_SSL_VERIFYHOST(3)* is 2, the SSL certificate provided by the DoH server must indicate that the server name is the same as the server name to which you meant to connect to, or the connection fails.

curl considers the DoH server the intended one when the Common Name field or a Subject Alternate Name field in the certificate matches the hostname in the DoH URL to which you told curl to connect.

When the *verify* value is set to 1L it is treated the same as 2L. However for consistency with the other *VERIFYHOST* options we suggest use 2 and not 1.

When the *verify* value is set to 0L, the connection succeeds regardless of the names used in the certificate. Use that ability with caution.

See also *CURLOPT_DOH_SSL_VERIFYPEER(3)* to verify the digital signature of the DoH server certificate.

DEFAULT

2

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");

        curl_easy_setopt(curl, CURLOPT_DOH_URL,
                        "https://cloudflare-dns.com/dns-query");

        /* Disable hostname verification of the DoH server */
        curl_easy_setopt(curl, CURLOPT_DOH_SSL_VERIFYHOST, 0L);

        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

}

AVAILABILITY

Added in curl 7.76.0

RETURN VALUE

curl_easy_setopt(3) returns a CURLcode indicating success or error.

CURL_OK (0) means everything was OK, non-zero means an error occurred, see *libcurl-errors(3)*.

SEE ALSO

CURLOPT_DOH_SSL_VERIFYPEER(3), CURLOPT_PROXY_SSL_VERIFYHOST(3), CUR-
LOPT_PROXY_SSL_VERIFYPEER(3), CURLOPT_SSL_VERIFYHOST(3), CUR-
LOPT_SSL_VERIFYPEER(3)