

NAME

CURLOPT_DOH_URL – provide the DNS-over-HTTPS URL

SYNOPSIS

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

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_DOH_URL, char *URL);
```

DESCRIPTION

Pass in a pointer to a *URL* for the DoH server to use for name resolving. The parameter should be a char pointer to a null-terminated string which must be a valid and correct HTTPS URL.

libcurl does not validate the syntax or use this variable until the transfer is issued. Even if you set a crazy value here, *curl_easy_setopt(3)* still returns *CURLE_OK*.

curl sends POST requests to the given DNS-over-HTTPS URL.

To find the DoH server itself, which might be specified using a name, libcurl uses the default name lookup function. You can bootstrap that by providing the address for the DoH server with *CURLOPT_RESOLVE(3)*.

The application does not have to keep the string around after setting this option.

Using this option multiple times makes the last set string override the previous ones. Set it to NULL to disable its use again.

INHERIT OPTIONS

DoH lookups use SSL and some SSL settings from your transfer are inherited, like *CURLOPT_SSL_CTX_FUNCTION(3)*.

The hostname and peer certificate verification settings are not inherited but can be controlled separately via *CURLOPT_DOH_SSL_VERIFYHOST(3)* and *CURLOPT_DOH_SSL_VERIFYPEER(3)*.

A set *CURLOPT_OPENSOCKETFUNCTION(3)* callback is not inherited.

KNOWN BUGS

Even when DoH is set to be used with this option, there are still some name resolves that are performed without it, using the default name resolver mechanism. This includes name resolves done for *CURLOPT_INTERFACE(3)*, *CURLOPT_FTPPORT(3)*, a proxy type set to **CURLPROXY_SOCKS4** or **CURLPROXY_SOCKS5** and probably some more.

DEFAULT

NULL

PROTOCOLS

This functionality affects all supported protocols

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://dns.example.com");
        result = curl_easy_perform(curl);
        curl_easy_cleanup(curl);
    }
}
```

AVAILABILITY

Added in curl 7.62.0

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)*.

Note that *curl_easy_setopt(3)* does immediately parse the given string so when given a bad DoH URL, libcurl might not detect the problem until it later tries to resolve a name with it.

SEE ALSO

CURLOPT_DNS_CACHE_TIMEOUT(3), CURLOPT_RESOLVE(3), CURLOPT_VERBOSE(3)