

NAME

CURLOPT_HSTSREADFUNCTION – read callback for HSTS hosts

SYNOPSIS

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

```
struct curl_hstsentry {  
    char *name;  
    size_t namelen;  
    unsigned int includeSubDomains:1;  
    char expire[18]; /* YYYYMMDD HH:MM:SS [null-terminated] */  
};
```

```
CURLSTSCode hstsread(CURL *easy, struct curl_hstsentry *sts, void *clientp);
```

```
CURLcode curl_easy_setopt(CURL *handle, CURLOPT_HSTSREADFUNCTION, hstsread);
```

DESCRIPTION

Pass a pointer to your callback function, as the prototype shows above.

This callback function gets called by libcurl repeatedly when it populates the in-memory HSTS cache.

Set the *clientp* argument with the *CURLOPT_HSTSREADDATA(3)* option or it is NULL.

When this callback is invoked, the *sts* pointer points to a populated struct: Copy the hostname to *name* (no longer than *namelen* bytes). Make it null-terminated. Set *includeSubDomains* to TRUE or FALSE. Set *expire* to a date stamp or a zero length string for *forever* (wrong date stamp format might cause the name to not get accepted)

The callback should return *CURLSTS_OK* if it returns a name and is prepared to be called again (for another host) or *CURLSTS_DONE* if it has no entry to return. It can also return *CURLSTS_FAIL* to signal error. Returning *CURLSTS_FAIL* stops the transfer from being performed and make *CURLE_ABORTED_BY_CALLBACK* get returned.

This option does not enable HSTS, you need to use *CURLOPT_HSTS_CTRL(3)* to do that.

The hostname provided to libcurl *should not* have a trailing dot nor leading dot.

DEFAULT

NULL – no callback.

PROTOCOLS

This functionality affects http only

EXAMPLE

```
struct priv {  
    void *custom;  
};  
  
static CURLSTSCode hsts_cb(CURL *easy, struct curl_hstsentry *sts,  
                           void *clientp)  
{  
    /* populate the struct as documented */  
    return CURLSTS_OK;  
}  
  
int main(void)
```

```
{
    CURL *curl = curl_easy_init();
    if(curl) {
        struct priv my_stuff;
        CURLcode result;

        curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");

        /* set HSTS read callback */
        curl_easy_setopt(curl, CURLOPT_HSTSREADFUNCTION, hsts_cb);

        /* pass in suitable argument to the callback */
        curl_easy_setopt(curl, CURLOPT_HSTSREADDATA, &my_stuff);

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

AVAILABILITY

Added in curl 7.74.0

RETURN VALUE

This returns CURLE_OK.

SEE ALSO

**CURLOPT_HSTS(3), CURLOPT_HSTSREADDATA(3), CURLOPT_HSTSWRITEFUNCTION(3),
CURLOPT_HSTS_CTRL(3)**