

NAME

CURLOPT_AWS_SIGV4 – V4 signature

SYNOPSIS

#include <curl/curl.h>

CURLcode curl_easy_setopt(CURL *handle, CURLOPT_AWS_SIGV4, char *param);

DESCRIPTION

Provides AWS V4 signature authentication on HTTP(S) header.

Pass a char pointer that is the collection of specific arguments are used for creating outgoing authentication headers. The format of the *param* option is:

provider1[:provider2[:region[:service]]]

provider1, provider2

The providers arguments are used for generating some authentication parameters such as "Algorithm", "date", "request type" and "signed headers".

region The argument is a geographic area of a resources collection. It is extracted from the hostname specified in the URL if omitted.

service The argument is a function provided by a cloud. It is extracted from the hostname specified in the URL if omitted.

NOTE: This call set *CURLOPT_HTTPAUTH(3)* to *CURLAUTH_AWS_SIGV4*. Calling *CURLOPT_HTTPAUTH(3)* with *CURLAUTH_AWS_SIGV4* is the same as calling this with "**aws:amz**" in parameter.

Example with "Test:Try", when curl uses the algorithm, it generates "**TEST-HMAC-SHA256**" for "Algorithm", "**x-try-date**" and "**X-Try-Date**" for "date", "**test4_request**" for "request type", "**Signed-Headers=content-type;host;x-try-date**" for "signed headers"

If you use "test", instead of "test:try", test is used for every generated string.

Setting *CURLOPT_HTTPAUTH(3)* with the *CURLAUTH_AWS_SIGV4* bit set is the same as setting this option with a "**aws:amz**" parameter.

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.

DEFAULT

NULL

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
    CURL *curl = curl_easy_init();

    if(curl) {
        CURLcode result;
        curl_easy_setopt(curl, CURLOPT_URL,
            "https://service.region.example.com/uri");
        curl_easy_setopt(curl, CURLOPT_AWS_SIGV4, "provider1:provider2");
```

```

/* service and region can also be set in CURLOPT_AWS_SIGV4 */
curl_easy_setopt(curl, CURLOPT_URL, "https://example.com/uri");
curl_easy_setopt(curl, CURLOPT_AWS_SIGV4,
    "provider1:provider2:region:service");

curl_easy_setopt(curl, CURLOPT_USERPWD, "MY_ACCESS_KEY:MY_SECRET_KEY");
result = curl_easy_perform(curl);
curl_easy_cleanup(curl);
}
}

```

NOTES

This option overrides the other auth types you might have set in *CURLOPT_HTTPAUTH(3)* which should be highlighted as this makes this auth method special. This method cannot be combined with other auth types.

A sha256 checksum of the request payload is used as input to the signature calculation. For POST requests, this is a checksum of the provided *CURLOPT_POSTFIELDS(3)*. Otherwise, it is the checksum of an empty buffer. For requests like PUT, you can provide your own checksum in an HTTP header named **x-provider2-content-sha256**.

For **aws:s3**, a **x-amz-content-sha256** header is added to every request if not already present. For s3 requests with unknown payload, this header takes the special value "UNSIGNED-PAYLOAD".

AVAILABILITY

Added in curl 7.75.0

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_HEADEROPT(3), **CURLOPT_HTTPAUTH(3)**, **CURLOPT_HTTPHEADER(3)**, **CURLOPT_PROXYAUTH(3)**