NAME

curl_easy_nextheader - get the next HTTP header

SYNOPSIS

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

struct curl_header *curl_easy_nextheader(CURL *easy,
                                         unsigned int origin,
                                         int request,
                                         struct curl_header *prev);
```

DESCRIPTION

This function lets an application iterate over all previously received HTTP headers.

The *origin* argument is for specifying which headers to receive, as a single HTTP transfer might provide headers from several different places and they may then have different importance to the user and headers using the same name might be used. The *origin* is a bitmask for what header sources you want. See the *curl_easy_header(3)* man page for the origin descriptions.

The *request* argument tells libcurl from which request you want headers from. A single transfer might consist of a series of HTTP requests and this argument lets you specify which particular individual request you want the headers from. 0 being the first request and then the number increases for further redirects or when multi-state authentication is used. Passing in -1 is a shortcut to "the last" request in the series, independently of the actual amount of requests used.

It is suggested that you pass in the same **origin** and **request** when iterating over a range of headers as changing the value mid-loop might give you unexpected results.

If *prev* is NULL, this function returns a pointer to the first header stored within the given scope (origin + request).

If *prev* is a pointer to a previously returned header struct, *curl_easy_nextheader(3)* returns a pointer the next header stored within the given scope. This way, an application can iterate over all available headers.

The memory for the struct this points to, is owned and managed by libcurl and is associated with the easy handle. Applications must copy the data if they want it to survive subsequent API calls or the life-time of the easy handle.

PROTOCOLS

This functionality affects http only

EXAMPLE

```
int main(void)
{
 struct curl_header *prev = NULL;
 struct curl_header *h;

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

  /* extract the normal headers from the first request */
  while((h = curl_easy_nextheader(curl, CURLH_HEADER, 0, prev))) {
   printf("%s: %s\n", h->name, h->value);
   prev = h;
  }

  /* extract the normal headers + 1xx + trailers from the last request */
  unsigned int origin = CURLH_HEADER| CURLH_1XX | CURLH_TRAILER;
  while((h = curl_easy_nextheader(curl, origin, -1, prev))) {
   printf("%s: %s\n", h->name, h->value);
   prev = h;
  }
 }
}
```

AVAILABILITY

Added in curl 7.83.0

RETURN VALUE

This function returns the next header, or NULL when there are no more (matching) headers or an error occurred.

If this function returns NULL when *prev* was set to NULL, then there are no headers available within the scope to return.

SEE ALSO

curl_easy_header(3), **curl_easy_perform**(3)