NAME

CURLINFO_EFFECTIVE_METHOD − last used HTTP request method

SYNOPSIS

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

CURLcode curl_easy_getinfo(CURL *handle, CURLINFO_EFFECTIVE_METHOD,
                           char **methodp);
```

DESCRIPTION

Pass in a pointer to a char pointer and get the last used effective HTTP method.

In cases when you have asked libcurl to follow redirects, the method may not be the same method the first request would use.

The **methodp** pointer is NULL or points to private memory. You **must not** free it. The memory gets freed automatically when you call *curl_easy_cleanup(3)* on the corresponding curl handle.

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://example.com");
  curl_easy_setopt(curl, CURLOPT_POSTFIELDS, "data");
  curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
  result = curl_easy_perform(curl);
  if(result == CURLE_OK) {
   char *method = NULL;
   curl_easy_getinfo(curl, CURLINFO_EFFECTIVE_METHOD, &method);
   if(method)
    printf("Redirected to method: %s\n", method);
  }
  curl_easy_cleanup(curl);
 }
}
```

AVAILABILITY

Added in curl 7.72.0

RETURN VALUE

curl_easy_getinfo(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_CUSTOMREQUEST(3), **CURLOPT_FOLLOWLOCATION**(3), **curl_easy_getinfo**(3), **curl_easy_setopt**(3)