NAME

curl_easy_cleanup - End a libcurl easy handle

SYNOPSIS

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

void curl_easy_cleanup(CURL *handle);
```

DESCRIPTION

This function is the opposite of *curl_easy_init(3)*. It closes down and frees all resources previously associated with this easy handle.

This call closes all connections this handle has used and possibly has kept open until now unless the easy handle was attached to a multi handle while doing the transfers. Do not call this function if you intend to transfer more files, reusing handles is a key to good performance with libcurl.

Occasionally you may get your progress callback or header callback called from within *curl_easy_cleanup(3)* (if previously set for the handle using *curl_easy_setopt(3)*). Like if libcurl decides to shut down the connection and the protocol is of a kind that requires a command/response sequence before disconnect. Examples of such protocols are FTP, POP3 and IMAP.

Any use of the easy **handle** after this function has been called and have returned, is illegal.

To close an easy handle that has been used with the multi interface, make sure to first call *curl_multi_remove_handle(3)* to remove it from the multi handle before it is closed.

Passing in a NULL pointer in *handle* makes this function return immediately with no action.

EXAMPLE

```
int main(void)
{
  CURL *curl = curl_easy_init();
  if(curl) {
    CURLcode res;
    curl_easy_setopt(curl, CURLOPT_URL, "https://example.com");
    res = curl_easy_perform(curl);
    if(res)
      printf("error: %s\n", curl_easy_strerror(res));
    curl_easy_cleanup(curl);
  }
}
```

AVAILABILITY

Added in 7.1

RETURN VALUE

None

SEE ALSO

curl_easy_duphandle(3), **curl_easy_init**(3), **curl_easy_reset**(3), **curl_multi_cleanup**(3), **curl_multi_remove_handle**(3)