

**NAME**

curl\_easy\_init - Start a libcurl easy session

**SYNOPSIS**

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

```
CURL *curl_easy_init( );
```

**DESCRIPTION**

This function must be the first function to call, and it returns a CURL easy handle that you must use as input to other functions in the easy interface. This call **MUST** have a corresponding call to *curl\_easy\_cleanup(3)* when the operation is complete.

If you did not already call *curl\_global\_init(3)*, *curl\_easy\_init(3)* does it automatically. This may be lethal in multi-threaded cases, since *curl\_global\_init(3)* is not thread-safe, and it may result in resource problems because there is no corresponding cleanup.

You are strongly advised to not allow this automatic behaviour, by calling *curl\_global\_init(3)* yourself properly. See the description in **libcurl(3)** of global environment requirements for details of how to use this function.

**RETURN VALUE**

If this function returns NULL, something went wrong and you cannot use the other curl functions.

**EXAMPLE**

```
CURL *curl = curl_easy_init();
if(curl) {
    CURLcode res;
    curl_easy_setopt(curl, CURLOPT_URL, "http://example.com");
    res = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
}
```

**SEE ALSO**

**curl\_easy\_cleanup(3)**, **curl\_global\_init(3)**, **curl\_easy\_reset(3)**, **curl\_easy\_perform(3)**