

## NAME

curl\_easy\_init - Start a libcurl easy session

## SYNOPSIS

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

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

## DESCRIPTION

This function allocates and returns a CURL easy handle. Such a handle is used 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.

The easy handle is used to hold and control a single network transfer. It is encouraged to reuse easy handles for repeated transfers.

An alternative way to get a new easy handle is to duplicate an already existing one with *curl\_easy\_duphandle(3)*, which has the upside that it gets all the options that were set in the source handle set in the new copy as well.

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

You are strongly advised to not allow this automatic behavior, 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.

## 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);
        curl_easy_cleanup(curl);
    }
}
```

curl\_easy\_init(3)

libcurl

curl\_easy\_init(3)

## **AVAILABILITY**

Always

## **RETURN VALUE**

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

## **SEE ALSO**

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