

NAME

curl_slist_append – add a string to an slist

SYNOPSIS

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

```
struct curl_slist *curl_slist_append(struct curl_slist *list,  
                                     const char *string);
```

DESCRIPTION

curl_slist_append(3) appends a string to a linked list of strings. The existing **list** should be passed as the first argument and the new list is returned from this function. Pass in NULL in the **list** argument to create a new list. The specified **string** has been appended when this function returns. *curl_slist_append(3)* copies the string. The **string** argument must be a valid string pointer and cannot be NULL.

The list should be freed (after usage) with *curl_slist_free_all(3)*. Its nodes and pointed content may not be altered outside this function.

PROTOCOLS

This functionality affects all supported protocols

EXAMPLE

```
int main(void)
{
    CURL *handle = curl_easy_init();
    struct curl_slist *slist = NULL;
    struct curl_slist *temp = NULL;

    slist = curl_slist_append(slist, "pragma:");

    if(!slist)
        return -1;

    temp = curl_slist_append(slist, "Accept:");

    if(!temp) {
        curl_slist_free_all(slist);
        return -1;
    }

    slist = temp;

    curl_easy_setopt(handle, CURLOPT_HTTPHEADER, slist);

    curl_easy_perform(handle);

    curl_slist_free_all(slist); /* free the list again */
}
```

AVAILABILITY

Added in curl 7.1

RETURN VALUE

A null pointer is returned if anything went wrong, otherwise the new list pointer is returned. To avoid overwriting an existing non-empty list on failure, the new list should be returned to a temporary variable which can be tested for NULL before updating the original list pointer.

SEE ALSO

curl_slist_free_all(3)