

## NAME

`ne_get_response_header`, `ne_response_header_iterate` - functions to access response headers

## SYNOPSIS

```
#include <ne_request.h>
```

```
const char *ne_get_response_header(ne_request *request, const char *name);
```

```
void *ne_response_header_iterate(ne_request *request, void *cursor, const char **name,  
                                const char **value);
```

## DESCRIPTION

To retrieve the value of a response header field, the `ne_get_response_header` function can be used, and is given the name of the header to return.

To iterate over all the response headers returned, the `ne_response_header_iterate` function can be used. This function takes a `cursor` parameter which should be NULL to retrieve the first header. The function stores the name and value of the next header header in the `name` and `value` parameters, and returns a new cursor pointer which can be passed to `ne_response_header_iterate` to retrieve the next header.

## RETURN VALUE

`ne_get_response_header` returns a string, or NULL if no header with that name was given. If used during request processing, the return value pointer is valid only until the next call to `ne_begin_request`, or else, until the request object is destroyed.

Likewise, the cursor, names, and values returned by `ne_response_header_iterate` are only valid until the next call to `ne_begin_request` or until the request object is destroyed.

## EXAMPLES

The following code will output the value of the Last-Modified header for a resource:

```
ne_request *req = ne_request_create(sess, "GET", "/foo.txt");  
if (ne_request_dispatch(req) == NE_OK) {  
    const char *mtime = ne_get_response_header(req, "Last-Modified");  
    if (mtime) {  
        printf("/foo.txt has last-modified value %s\n", mtime);  
    }  
}  
ne_request_destroy(req);
```

NE\_GET\_RESPONSE\_HEAD(3)

neon API reference

NE\_GET\_RESPONSE\_HEAD(3)

## SEE ALSO

ne\_request\_create, ne\_request\_destroy.

## AUTHOR

**Joe Orton** <neon@lists.manyfish.co.uk>

Author.

## COPYRIGHT