This repository Search | Explore | Gist | Blog | Help | itpp16

openresty / **headers-more-nginx-module**

Watch 32 | Star 260 | Fork 33

Set, add, and clear arbitrary output headers http://wiki.nginx.org/NginxHttpHeadersMoreModule

195 commits | **1** branch | **39** releases | **3** contributors

branch: **master** | **headers-more-nginx-module** / +

updated doc to reflect recent changes.

agentzh authored on Dec 9, 2014 | latest commit a7f81f20be

doc	updated doc to reflect recent changes.	2 months ago
src	style: fixed the coding style of labels.	6 months ago
t	bugfix: more_set_input_headers did not completely override the existi…	a year ago
util	bugfix: modifying the X-Real-IP request header via more_set_input_hea…	2 years ago
.gitignore	updated .gitignore a bit.	2 years ago
README.markdown	updated doc to reflect recent changes.	2 months ago
config	renamed the source file names a bit.	4 years ago
valgrind.suppress	suppressed a valgrind false positive in libdl.	11 months ago

Code | Issues 7 | Pull Requests 0 | Wiki | Pulse | Graphs

HTTPS clone URL

https://github.com/c

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop | Download ZIP

README.markdown

Name

ngx_headers_more - Set and clear input and output headers...more than "add"!

This module is not distributed with the Nginx source. See the installation instructions.

Table of Contents

Version

This document describes headers-more-nginx-module v0.25 released on 10 January 2014.

Synopsis

```
# set the Server output header
more_set_headers 'Server: my-server';

# set and clear output headers
location /bar {
    more_set_headers 'X-MyHeader: blah' 'X-MyHeader2: foo';
    more_set_headers -t 'text/plain text/css' 'Content-Type: text/foo';
    more_set_headers -s '400 404 500 503' -s 413 'Foo: Bar';
    more_clear_headers 'Content-Type';

    # your proxy_pass/memcached_pass/or any other config goes here...
}

# set output headers
location /type {
    more_set_headers 'Content-Type: text/plain';
    # ...
}

# set input headers
location /foo {
    set $my_host 'my dog';
    more_set_input_headers 'Host: $my_host';
    more_set_input_headers -t 'text/plain' 'X-Foo: bah';

    # now $host and $http_host have their new values...
    # ...
}

# replace input header X-Foo *only* if it already exists
more_set_input_headers -r 'X-Foo: howdy';
```

Description

This module allows you to add, set, or clear any output or input header that you specify.

This is an enhanced version of the standard headers module because it provides more utilities like resetting or clearing "builtin headers" like `Content-Type`, `Content-Length`, and `Server`.

It also allows you to specify an optional HTTP status code criteria using the `-s` option and an optional content type criteria using the `-t` option while modifying the output headers with the more_set_headers and more_clear_headers directives. For example,

```
more_set_headers -s 404 -t 'text/html' 'X-Foo: Bar';
```

Input headers can be modified as well. For example

```
location /foo {
    more_set_input_headers 'Host: foo' 'User-Agent: faked';
    # now $host, $http_host, $user_agent, and
    #   $http_user_agent all have their new values.
}
```

The option `-t` is also available in the more_set_input_headers and more_clear_input_headers directives (for request header filtering) while the `-s` option is not allowed.

Unlike the standard headers module, this module's directives will by default apply to all the status codes, including `4xx` and `5xx`.

Back to TOC

Directives

Back to TOC

more_set_headers

syntax: *more_set_headers [-t <content-type list>]... [-s <status-code list>]... <new-header>...*

default: *no*

context: *http, server, location, location if*

phase: *output-header-filter*

Replaces (if any) or adds (if not any) the specified output headers when the response status code matches the codes specified by the `-s` option *AND* the response content type matches the types specified by the `-t` option.

If either `-s` or `-t` is not specified or has an empty list value, then no match is required. Therefore, the following directive set the `Server` output header to the custom value for *any* status code and *any* content type:

```
 more_set_headers    "Server: my_server";
```

Existing response headers with the same name are always overridden. If you want to add headers incrementally, use the standard add_header directive instead.

A single directive can set/add multiple output headers. For example

```
more_set_headers 'Foo: bar' 'Baz: bah';
```

Multiple occurrences of the options are allowed in a single directive. Their values will be merged together. For instance

```
more_set_headers -s 404 -s '500 503' 'Foo: bar';
```

is equivalent to

```
more_set_headers -s '404 500 503' 'Foo: bar';
```

The new header should be the one of the forms:

1. `Name: Value`
2. `Name:`
3. `Name`

The last two effectively clear the value of the header `Name`.

Nginx variables are allowed in header values. For example:

```
 set $my_var "dog";
 more_set_headers "Server: $my_var";
```

But variables won't work in header keys due to performance considerations.

Multiple set/clear header directives are allowed in a single location, and they're executed sequentially.

Directives inherited from an upper level scope (say, http block or server blocks) are executed before the directives in the location block.

Note that although `more_set_headers` is allowed in *location* if blocks, it is *not* allowed in the *server* if blocks, as in

```
? # This is NOT allowed!
? server {
?     if ($args ~ 'download') {
?         more_set_headers 'Foo: Bar';
?     }
?     ...
? }
```

Behind the scene, use of this directive and its friend more_clear_headers will (lazily) register an ouput header filter that modifies `r->headers_out` the way you specify.

Back to TOC

more_clear_headers

syntax: *more_clear_headers [-t <content-type list>]... [-s <status-code list>]... <new-header>...*

default: *no*

context: *http, server, location, location if*

phase: *output-header-filter*

Clears the specified output headers.

In fact,

```
    more_clear_headers -s 404 -t 'text/plain' Foo Baz;
```

is exactly equivalent to

```
    more_set_headers -s 404 -t 'text/plain' "Foo: " "Baz: ";
```

or

```
    more_set_headers -s 404 -t 'text/plain' Foo Baz
```

See more_set_headers for more details.

Wildcard `*` can also be used to specify a header name pattern. For example, the following directive effectively clears *any* output headers starting by "`X-Hidden-`":

```
more_clear_headers 'X-Hidden-*';
```

The `*` wildcard support was first introduced in v0.09.

Back to TOC

more_set_input_headers

syntax: *more_set_input_headers [-r] [-t <content-type list>]... <new-header>...*

default: *no*

context: *http, server, location, location if*

phase: *rewrite tail*

Very much like more_set_headers except that it operates on input headers (or request headers) and it only supports the `-t` option.

Note that using the `-t` option in this directive means filtering by the `Content-Type` *request* header, rather than the response header.

Behind the scene, use of this directive and its friend more_clear_input_headers will (lazily) register a `rewrite phase` handler that modifies `r->headers_in` the way you specify. Note that it always run at the *end* of the `rewrite` so that it runs *after* the standard rewrite module and works in subrequests as well.

If the `-r` option is specified, then the headers will be replaced to the new values *only if* they already exist.

Back to TOC

more_clear_input_headers

syntax: *more_clear_input_headers [-t <content-type list>]... <new-header>...*

default: *no*

context: *http, server, location, location if*

phase: *rewrite tail*

Clears the specified input headers.

In fact,

```
    more_clear_input_headers -s 404 -t 'text/plain' Foo Baz;
```

is exactly equivalent to

```
more_set_input_headers -s 404 -t 'text/plain' "Foo: " "Baz: ";
```

or

```
more_set_input_headers -s 404 -t 'text/plain' Foo Baz
```

See more_set_input_headers for more details.

Back to TOC

Limitations

- Unlike the standard headers module, this module does not automatically take care of the constraint among the `Expires`, `Cache-Control`, and `Last-Modified` headers. You have to get them right yourself or use the headers module together with this module.
- You cannot remove the `Connection` response header using this module because the `Connection` response header is generated by the standard `ngx_http_header_filter_module` in the Nginx core, whose output header filter runs always *after* the filter of this module. The only way to actually remove the `Connection` header is to patch the Nginx core, that is, editing the C function `ngx_http_header_filter` in the `src/http/ngx_http_header_filter_module.c` file.

Back to TOC

Installation

Grab the nginx source code from nginx.org, for example, the version 1.7.7 (see nginx compatibility), and then build the source with this module:

```
wget 'http://nginx.org/download/nginx-1.7.7.tar.gz'
tar -xzvf nginx-1.7.7.tar.gz
cd nginx-1.7.7/

# Here we assume you would install you nginx under /opt/nginx/.
./configure --prefix=/opt/nginx \
    --add-module=/path/to/headers-more-nginx-module

make
make install
```

Download the latest version of the release tarball of this module from headers-more-nginx-module file list.

Also, this module is included and enabled by default in the ngx_openresty bundle.

Back to TOC

Compatibility

The following versions of Nginx should work with this module:

- **1.7.x** (last tested: 1.7.7)
- **1.6.x** (last tested: 1.6.2)
- **1.5.x** (last tested: 1.5.8)
- **1.4.x** (last tested: 1.4.4)
- **1.3.x** (last tested: 1.3.7)
- **1.2.x** (last tested: 1.2.9)

- **1.1.x** (last tested: 1.1.5)
- **1.0.x** (last tested: 1.0.11)
- **0.9.x** (last tested: 0.9.4)
- **0.8.x** (last tested: 0.8.54)
- **0.7.x >= 0.7.44** (last tested: 0.7.68)

Earlier versions of Nginx like 0.6.x and 0.5.x will *not* work.

If you find that any particular version of Nginx above 0.7.44 does not work with this module, please consider reporting a bug.

Back to TOC

Community

Back to TOC

English Mailing List

The openresty-en mailing list is for English speakers.

Back to TOC

Chinese Mailing List

The openresty mailing list is for Chinese speakers.

Back to TOC

Bugs and Patches

Please submit bug reports, wishlists, or patches by

1. creating a ticket on the GitHub Issue Tracker,
2. or posting to the OpenResty community.

Back to TOC

Source Repository

Available on github at agentzh/headers-more-nginx-module.

Back to TOC

Changes

The changes of every release of this module can be obtained from the ngx_openresty bundle's change logs:

http://openresty.org/#Changes

Back to TOC

Test Suite

This module comes with a Perl-driven test suite. The test cases are declarative too. Thanks to the

Test::Nginx module in the Perl world.

To run it on your side:

```
$ PATH=/path/to/your/nginx-with-headers-more-module:$PATH prove -r t
```

To run the test suite with valgrind's memcheck, use the following commands:

```
$ export PATH=/path/to/your/nginx-with-headers-more-module:$PATH
$ TEST_NGINX_USE_VALGRIND=1 prove -r t
```

You need to terminate any Nginx processes before running the test suite if you have changed the Nginx server binary.

Because a single nginx server (by default, `localhost:1984`) is used across all the test scripts (`.t` files), it's meaningless to run the test suite in parallel by specifying `-jN` when invoking the `prove` utility.

Some parts of the test suite requires modules proxy, rewrite, and echo to be enabled as well when building Nginx.

Back to TOC

TODO

- Support variables in new headers' keys.

Back to TOC

Getting involved

You'll be very welcomed to submit patches to the author or just ask for a commit bit to the source repository on GitHub.

Back to TOC

Authors

- Yichun "agentzh" Zhang (章亦春) <*agentzh@gmail.com*>, CloudFlare Inc.
- Bernd Dorn (http://www.lovelysystems.com/)

This wiki page is also maintained by the author himself, and everybody is encouraged to improve this page as well.

Back to TOC

Copyright & License

The code base is borrowed directly from the standard headers module in Nginx 0.8.24. This part of code is copyrighted by Igor Sysoev.

Back to TOC

See Also

- The original thread on the Nginx mailing list that inspires this module's development: "A question about add_header replication".
- The orginal announcement thread on the Nginx mailing list: "The "headers_more" module: Set and clear output headers...more than 'add'!".
- The original blog post about this module's initial development.
- The echo module for Nginx module's automated testing.
- The standard headers module.