Apache Module mod_headers

Description:	Customization of HTTP request and response headers
Status:	Extension
Module Identifier:	headers_module
Source File:	mod_headers.c
Compatibility:	`RequestHeader` is available only in Apache 2.0

Summary

This module provides directives to control and modify HTTP request and response headers. Headers can be merged, replaced or removed.

Topics

Directives

Order of Processing

The directives provided by `mod_headers` can occur almost anywhere within the server configuration. They are valid in the main server config and virtual host sections, inside `<Directory>`, `<Location>` and `<Files>` sections, and within `.htaccess` files.

The directives are processed in the following order:

1. main server
2. virtual host
3. `<Directory>` sections and `.htaccess`
4. `<Files>`
5. `<Location>`

Order is important. These two headers have a different effect if reversed:

```
RequestHeader append MirrorID "mirror 12"
RequestHeader unset MirrorID
```

This way round, the `MirrorID` header is not set. If reversed, the MirrorID header is set to "mirror 12".

Examples

1. Copy all request headers that begin with "TS" to the response headers:

   ```
   Header echo ^TS
   ```

2. Add a header, `MyHeader`, to the response including a timestamp for when the request was received and how long it took to begin serving the request. This header can be used by the client

to intuit load on the server or in isolating bottlenecks between the client and the server.

```
Header add MyHeader "%D %t"
```

results in this header being added to the response:

```
MyHeader: D=3775428 t=991424704447256
```

3. Say hello to Joe

```
Header add MyHeader "Hello Joe. It took %D microseconds \
for Apache to serve this request."
```

results in this header being added to the response:

```
MyHeader: Hello Joe. It took D=3775428 microseconds for Apache to
serve this request.
```

4. Conditionally send `MyHeader` on the response if and only if header "MyRequestHeader" is present on the request. This is useful for constructing headers in response to some client stimulus. Note that this example requires the services of the `mod_setenvif` module.

```
SetEnvIf MyRequestHeader value HAVE_MyRequestHeader
Header add MyHeader "%D %t mytext" env=HAVE_MyRequestHeader
```

If the header `MyRequestHeader: value` is present on the HTTP request, the response will contain the following header:

```
MyHeader: D=3775428 t=991424704447256 mytext
```

Header Directive

Description:	Configure HTTP response headers

Syntax:	`Header set|append|add|unset|echo` *`header`* `[`*`value`*
	`[env=[!]`*`variable`*`]]`

Context:	server config, virtual host, directory, .htaccess
Override:	FileInfo
Status:	Extension
Module:	mod_headers

This directive can replace, merge or remove HTTP response headers. The header is modified just after the content handler and output filters are run, allowing outgoing headers to be modified. The action it performs is determined by the first argument. This can be one of the following values:

`set`
The response header is set, replacing any previous header with this name. The *value* may be a format string.

`append`
The response header is appended to any existing header of the same name. When a new value is merged onto an existing header it is separated from the existing header with a comma. This is the HTTP standard way of giving a header multiple values.

add
: The response header is added to the existing set of headers, even if this header already exists. This can result in two (or more) headers having the same name. This can lead to unforeseen consequences, and in general "append" should be used instead.

unset
: The response header of this name is removed, if it exists. If there are multiple headers of the same name, all will be removed.

echo
: Request headers with this name are echoed back in the response headers. *header* may be a regular expression.

This argument is followed by a *header* name, which can include the final colon, but it is not required. Case is ignored for `set`, `append`, `add` and `unset`. The *header* name for `echo` is case sensitive and may be a regular expression.

For `add`, `append` and `set` a *value* is specified as the third argument. If *value* contains spaces, it should be surrounded by doublequotes. *value* may be a character string, a string containing format specifiers or a combination of both. The following format specifiers are supported in *value*:

`%t`	The time the request was received in Universal Coordinated Time since the epoch (Jan. 1, 1970) measured in microseconds. The value is preceded by `t=`.
`%D`	The time from when the request was received to the time the headers are sent on the wire. This is a measure of the duration of the request. The value is preceded by `D=`.
`%{FOOBAR}e`	The contents of the environment variable[1] `FOOBAR`.

When the `Header` directive is used with the `add`, `append`, or `set` argument, a fourth argument may be used to specify conditions under which the action will be taken. If the environment variable[1] specified in the `env=...` argument exists (or if the environment variable does not exist and `env=!...` is specified) then the action specified by the `Header` directive will take effect. Otherwise, the directive will have no effect on the request.

The `Header` directives are processed just before the response is sent to the network. These means that it is possible to set and/or override most headers, except for those headers added by the header filter.

RequestHeader Directive

Description:	Configure HTTP request headers
Syntax:	`RequestHeader set|append|add|unset` *`header`* `[`*`value`*`]`
Context:	server config, virtual host, directory, .htaccess
Override:	FileInfo
Status:	Extension
Module:	mod_headers

This directive can replace, merge or remove HTTP request headers. The header is modified just before the content handler is run, allowing incoming headers to be modified. The action it performs is determined by the first argument. This can be one of the following values:

set
: The request header is set, replacing any previous header with this name

append

The request header is appended to any existing header of the same name. When a new value is merged onto an existing header it is separated from the existing header with a comma. This is the HTTP standard way of giving a header multiple values.

`add`

The request header is added to the existing set of headers, even if this header already exists. This can result in two (or more) headers having the same name. This can lead to unforeseen consequences, and in general `append` should be used instead.

`unset`

The request header of this name is removed, if it exists. If there are multiple headers of the same name, all will be removed.

This argument is followed by a header name, which can include the final colon, but it is not required. Case is ignored. For `add`, `append` and `set` a *value* is given as the third argument. If *value* contains spaces, it should be surrounded by double quotes. For unset, no *value* should be given.

The `RequestHeader` directive is processed just before the request is run by its handler in the fixup phase. This should allow headers generated by the browser, or by Apache input filters to be overridden or modified.

URI References

[1] http://httpd.apache.org/docs-2.1/env.html