HTTPHeader

Last updated: Tue, 05 Jan 2021 14:15:08 GMT

Learn how to use the HTTPHeader config.txt directive to configure EZproxy to pass through custom HTTP headers created by browser applications to pass information to EZproxy. HTTPHeader allows EZproxy administrators to configure EZproxy to pass through custom HTTP headers created by browser applications to pass information to EZproxy. Without the HTTPHeader directive, EZproxy will ignore these custom headers and allow only those headers it has been explicitly configured to handle.

HTTPHeader allows users to specify additional response headers to be sent from EZproxy when the program is responding to a request and acting as its own server, for example, during login or presentation of /admin pages.

HTTPHeader is a global directive that affects the non-database oriented operation of EZproxy. These directives cannot be used to override or replace any of the headers already sent by EZproxy. Care should be taken when choosing to add additional headers that are sent back from EZproxy.

Syntax

The HTTPHeader directive takes the following form:

HTTPHeader -server header expression

Any entries using the standard form of HTTPHeader wildheader from before V6.1 will be read as:

HTTPHeader -request -global CustomHeader

Qualifiers

The following qualifiers can be used with the HTTPHeader directive to customize how EZproxy processes headers.

<table>
<thead>
<tr>
<th>QUALIFIER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>-global</td>
<td>Apply this directive to all database stanzas except those that contain HTTPHeader -ignoreGlobal.</td>
</tr>
<tr>
<td>-ignoreGlobal</td>
<td>Within the current database stanza, ignore all HTTPHeader -global directives defined in config.txt. The HTTPHeader directive with this qualifier must be placed AFTER the Title directive to apply to this stanza.</td>
</tr>
<tr>
<td>-direction: Use these options to designate how EZproxy should process the header.</td>
<td></td>
</tr>
<tr>
<td>-request</td>
<td>Designates the header is in the request from the user’s browser to the content provider.</td>
</tr>
<tr>
<td>QUALIFIER</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-response</td>
<td>Designates the header is in the response from the content provider to the user's browser.</td>
</tr>
<tr>
<td>-method: Use these options to designate how EZproxy should process the header</td>
<td></td>
</tr>
<tr>
<td>-process</td>
<td>Process the header by allowing EZproxy to perform standard processing on the header's value or allow a header that EZproxy would normally block to pass through.</td>
</tr>
<tr>
<td>-block</td>
<td>Block the header in its value</td>
</tr>
<tr>
<td>-rewrite</td>
<td>Treat the value of the header as a URL and rewrite it into EZproxy form (this is typically used with -response).</td>
</tr>
<tr>
<td>-unrewrite</td>
<td>Treat the value of the header as a URL that is rewritten in EZproxy form that needs to be unrewritten into its original form (typically used with -request).</td>
</tr>
<tr>
<td>-edit</td>
<td>Provide the header as hh:header and its value as hh:value to the user-specified expression; if the returned value is non-empty, send the header with that value; if the returned value is empty, suppress the header.</td>
</tr>
<tr>
<td>header</td>
<td>HTTP header to be affected.</td>
</tr>
<tr>
<td>expression</td>
<td>Provide the header as hh:header to the user-specified expression; if the returned value is non-empty, send the header with that value; if the returned value is empty, suppress the header.</td>
</tr>
</tbody>
</table>

**Example**

To include the X-Frame-Options header with a value of DENY:

```
HTTPHeader -server X-Frame-Options "DENY"
```

This combination would prevent the login page from being presented inside a frame.

The HTTPHeader allows you to apply the HTTPHeader directive selectively to database stanzas. For example, the header specified in the following stanza will only be applied to this single resource since it is placed after the Title directive and does not contain the -global qualifier:

```
Title First Database
HTTPHeader -request CustomHeader
```
The following configuration will apply the first HTTPHeader directive to the first and third database stanzas as well as any others in config.txt, but not the middle stanza because it includes the HTTPHeader -ignoreGlobal directive after the Title directive. Using the -ignoreGlobal qualifier with this directive will cause this stanza to ignore all global HTTPHeader directives that are located anywhere in config.txt.

```
HTTPHeader -global SpecialHeader
Title Science Database
URL http://www.sciencedb.com
DJ sciencedb.com

Title History Database
HTTPHeader -ignoreGlobal SpecialHeader
URL http://www.historydb.com
DJ historydb.com

Title Literature Database
URL http://www.litdb.com
DJ litdb.com
```

Related directives

- CompareIP(str1,str2)
- RewriteURL(str)
- RequestURL(str)
- UnwriteURL(str)