



IBM Software Group

# Breakdown - HTTP Response Headers

*Advisory Software Engineer – Robert L Boretti Jr*  
*WebSphere® Application Server L2 Support*



WebSphere® Support Technical Exchange



# Agenda

- RFC 2616 section 14 - **Header Field Definitions**
- Common **Response** Header Fields (HTTP/1.1) observed in WebSphere Application Server environments (e.g Server, Set-Cookie Content-Length, Location, etc..)
  - ▶ Specifically
    - Description of common *Response Header Fields*
    - Examples
  - ▶ Where they are logged
    - Packet Traces
    - Web server access logs
    - HTTP plug-in Trace log
    - WebSphere Application Server Trace log



# RFC 2616 section 14

- Google **key word** search “*HTTP response headers*”
- Direct link to *section 14* of the *RFC 2616*
  - ▶ <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>



# Common Response Headers

- **Content-Length** - The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs. Used to notify the client of how much content to expect to be sent by the server.

*example:*

**Content-Length: 3495**

- **Content-Type** - The Content-Type entity-header field indicates the media type of the entity-body sent to the recipient. Again, used to notify the client of what type of content the server is sending.

*example:*

**Content-Type: text/html; charset=ISO-8859-4**

- **Connection** - The Connection general-header field allows the sender to specify options that are desired for that particular connection.

*example:*

**Connection: close**

*Note: When this header is sent in a response back to the client, the connection on the server side is closed after the last byte of data has been sent by the server.*



# Common Response Headers

- **Cache-Control** - The Cache-Control general-header field is used to specify directives that **MUST** be obeyed by all caching mechanisms along the request/response chain. The directives specify behavior intended to prevent caches from adversely interfering with the request or response.

*example:*

**Cache-Control: no-cache**

- **Date** - The Date general-header field represents the date and time at which the message was originated, having the same semantics as orig-date in RFC 822. The field value is an HTTP-date, as described in section 3.3.1 in RFC 2616; it **MUST** be sent in RFC 1123 [8]-date format.

*example:*

**Date: Tue, 22 Jul 2008 19:12:37 GMT**

## **For more information..**

*RFC 822*

<http://www.w3.org/Protocols/rfc822/>

*RFC 2616*

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

*RFC 1123*

<http://www.ietf.org/rfc/rfc1123.txt>



# Common Response Headers

- **Expires** - The Expires entity-header field gives the date/time after which the response is considered stale.

*example:*

Expires: Thu, 01 Dec 1994 16:00:00 GMT

- \*If a response includes a Cache-Control field with the max-age directive (see section 14.9.3 of RFC 2616), that directive overrides the Expires field.*
- \*To mark a response as "already expired," an origin server sends an Expires date that is equal to the Date header value.*
- \*To mark a response as "never expires," an origin server sends an Expires date approximately one year from the time the response is sent.*
- \*The presence of an Expires header field with a date value of some time in the future on a response that otherwise would by default be non-cacheable indicates that the response is cacheable, unless indicated otherwise by a Cache-Control header field.*



# Common Response Headers

- **Location** - The Location response-header field is used to redirect the recipient to a location other than the Request-URI for completion of the request or identification of a new resource.

*example:*

Location: <https://www.ibm.com/login.jsp>

- **Content-Encoding** - The Content-Encoding entity-header field is used as a modifier to the media-type. When present, its value indicates what additional content codings have been applied to the entity-body, and thus what decoding mechanisms must be applied in order to obtain the media-type referenced by the Content-Type header field. Content-Encoding is primarily used to allow a document to be compressed without losing the identity of its underlying media type.

*example:*

Content-Encoding: gzip



# Common Response Headers

- **Set-Cookie** - The Set-Cookie header is included with a response to notify the client to store the cookie. The client should include the cookie in the *Cookie* request header when making additional HTTP requests to the server.

*example:*

[Set-Cookie: JSESSIONID=0000CARXI2bhwMkByU9IJkth\\_ns:12d5oviv](#)

- **Server** - The Server response-header field contains information about the software used by the origin server to handle the request.

*example:*

[Server: WebSphere Application Server/6.0](#)

- **Last-Modified** - The Last-Modified entity-header field indicates the date and time at which the origin server believes the variant was last modified. Typically included with a 304 status code response

*example:*

[Last-Modified: Sun, 14 May 2006 21:35:56 GMT](#)





# Common Response Headers

- **Transfer-Encoding** - The Transfer-Encoding general-header field indicates what (if any) type of transformation has been applied to the message body in order to safely transfer it between the sender and the recipient.

*example:*

[Transfer-Encoding: chunked](#)

- **WWW-Authenticate** - The WWW-Authenticate response-header field **MUST** be included in 401 (Unauthorized) response messages. The field value consists of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the Request-URI.

*example:*

[WWW-Authenticate: Basic realm=www.ibm.com](#)



# Where Response Headers are logged

- **Packet Traces** (trace.pcap)
  - no example provided due to customer confidentiality
- **Web Server access log** (access\_log)
  - An outgoing HTTP response header content can be logged in the IBM HTTP server's access\_log using a custom log format, `%{headername}o` and defining this using the LogFormat directive in the httpd.conf.

*example:*

```
LogFormat "%v %h %l %u %t \"%r\" %{Content-Type}o %>s %b"
```

*example of access\_log output:*

```
127.0.0.1 - - [26/Sep/2008:17:31:00 -0400] "GET / HTTP/1.1"
```

```
text/html 200 3183
```



# Where Response Headers are logged

- **HTTP plug-in log** (http\_plugin.log)

When loglevel in the plugin-cfg.xml is set to Trace, HTTP response headers which are received by the HTTP plug-in from the WebSphere Application Server are logged in the plugin log.

*example:*

```
<Log LogLevel="Trace" Name="c:\temp\logs\http_plugin.log"/>
```

*example of http\_plugin.log output:*

```
DETAIL: HTTP/1.1 200 OK
DETAIL: Content-Type: text/html;charset=ISO-8859-1
DETAIL: Content-Language: en
DETAIL: Set-Cookie: JSESSIONID=000052zU_s2N8n8dKGdiGXIONa1:12d5ovmq8; Path=/
DETAIL: Transfer-Encoding: chunked
DETAIL: Date: Tue, 22 Jul 2008 19:12:37 GMT
DETAIL: Server: WebSphere Application Server/6.1
DETAIL: Expires: Thu, 01 Dec 1994 16:00:00 GMT
DETAIL: Cache-Control: no-cache="set-cookie, set-cookie2"
```

# Where Response Headers are logged

- **WebSphere Application Server trace log** (trace.log)

*example:*

```
[9/19/08 8:59:39:112 EDT] 00000211 SRTServletRes 3  setHeader name --> Location value -->
https://www.ibm.com/ : com.ibm.ws.webcontainer.srt.SRTServletResponse@7aba5a53
[9/19/08 8:59:39:112 EDT] 00000211 SRTServletRes 3  setHeader name --> Location value -->
https:// www.ibm.com/ checkInclude --> true :
com.ibm.ws.webcontainer.srt.SRTServletResponse@7aba5a53
[9/19/08 8:59:39:113 EDT] 00000211 SRTServletReq 3  getWebAppDispatcherContext
[com.ibm.ws.webcontainer.srt.SRTServletRequest@496889eb]
[9/19/08 8:59:39:113 EDT] 00000211 BNFHeadersImp 3  setHeader(s,s): Location
[9/19/08 8:59:39:113 EDT] 00000211 BinaryHeaderN 3  Char value [Location] matches key: Key:
Location Ordinal: 34
[9/19/08 8:59:39:113 EDT] 00000211 BNFHeadersImp 3  setHeader(h,s): Location
[9/19/08 8:59:39:114 EDT] 00000211 BNFHeadersImp 3  Adding known header [Location] with
value [https:// www.ibm.com/ ]
```

Note: Recommended trace string to allow this level of logging in trace.log

```
*=info:com.ibm.ws.webcontainer.*=all:com.ibm.wsspi.webcontainer.*=all:HTTPChannel=all:Generi
cBNF=all
```

# Reference

- For more information on **HTTP Response Headers** and **Hypertext Transfer Protocol -- HTTP/1.1**

<http://www.w3.org/Protocols/rfc2616/rfc2616.html>



# Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Learn about other upcoming webcasts, conferences and events:  
[http://www.ibm.com/software/websphere/events\\_1.html](http://www.ibm.com/software/websphere/events_1.html)
- Join the Global WebSphere User Group Community: <http://www.websphere.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>



# Questions and Answers

