Breakdown - HTTP Request Headers

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

Agenda

- RFC 2616 section 14 - **Header Field Definitions**
- Common **Request** Header Fields (HTTP/1.1) observed in WebSphere Application Server environments (e.g *Host*, *Cookie*, *Accept*, *User-Agent*, etc..)
  - Specifically
    - Overview of common *Request Methods* (GET, HEAD, POST)
    - Description of common *Request 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 Request headers”*
- Direct link to *section 14* of the *RFC 2616*
    - http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

Common Request Methods

- **GET** -The GET method means *retrieve* whatever information is identified by the Request-URI. In simple terms, when a client issues a GET it is asking the server for something.

- **HEAD** - The HEAD method is identical to GET except that the server MUST NOT return a *message-body* in the response. In simple terms, when a client issues a HEAD request it typically is looking to receive the response *status code (e.g 200, 304, etc..)* only and not the actual body content.

  *Many Load balancers are configured to issue HEAD requests to check the health of a server.

- **POST** - The POST method is used to request that the origin server *accept the entity* enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. In simple terms, when a client issues a POST request it is sending data to the server (e.g.. uploading a file, submitting user information, credit card data, etc)

Common Request Headers

- **Accept** - The Accept request-header field can be used to specify certain media types which are acceptable for the response.

  example:

  Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*

- **Accept-Encoding** - The Accept-Encoding request-header field is similar to Accept, but restricts the content-codings (section 3.5 of RFC 2616) that are acceptable in the response. Typically, used to inform the server that the client will accept a response that is *compressed*

  example:

  Accept-encoding: compress, gzip

  - **Accept-Language** - The Accept-Language request-header field is similar to Accept, but restricts the set of natural languages that are preferred as a response to the request. Language tags are defined in section 3.10 of RFC 2616.

    example:

    Accept-language: en-us

Common Request Headers

- **Authorization** - A client that wishes to authenticate itself with a server-- usually, but not necessarily, after receiving a 401 response--does so by including an Authorization request-header field with the request. The Authorization field value consists of credentials containing the authentication information of the client for the realm of the resource being requested.

  example:

  Authorization: Basic <username:password>

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

    example:

    Connection: close

  - **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. Typically seen for POST requests.

    example:

    Cache-Control: no-cache

Common Request Headers

- **Content-Length** - The Content-Length entity-header field indicates the size of the entity-body, in decimal number of OCTETs. Typically seen for POST requests to notify the server of how much content the client is sending.

  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, typically seen for POST requests to notify the server of what type of content the client is sending.

  example:

  Content-Type: text/html; charset=ISO-8859-4
- **Host** - The Host request-header field specifies the Internet host and port number of the resource being requested, as obtained from the original URI given by the user or referring resource

  example:

  Host: www.ibm.com

Common Request Headers

- **If-Modified-Since** - The If-Modified-Since request-header field is used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, an entity will not be returned from the server; instead, a 304 (not modified) response will be returned without any message-body.
  example:
  If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT
- **Referer** - The Referer request-header field allows the client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained.
  example:
  Referer: http://www.ibm.com/us/index.html
- **User-Agent** - The User-Agent request-header field contains information about the client originating the request.
  example:
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)

Common Request Headers

- **Expect** - The Expect request-header field is used to indicate that particular server behaviors are required by the client. The HTTP plug-in will always send an Expect: 100-Continue header when issuing a POST request to a backend WebSphere Application Server.

  example:

  Expect: 100-Continue

  Note: The purpose of the 100 (Continue) status (see section 10.1.1 in RFC 2616) is to allow a client that is sending a request message with a request body to determine if the origin server is willing to accept the request (based on the request headers) before the client sends the request body.
- **Cookie** - The Cookie header is included by the client with an HTTP request sent to a server

  example:

  Cookie: JSESSIONID=0000CARXl2bhwMkByU9lJkth_ns:12d5ovivu

Common Plug-in Request Headers

- **Surrogate-Capability** - When the ESI processor is enabled in the plug-in and a cache miss occurs, the Surrogate-Capabilities header is added to the request and the request is forwarded to the WebSphere Application Server.

  example:

  Surrogate-Capability: WS-ESI="ESI/1.0+"

- **$WSAT** - The Auth Type that is being used to make this request.

  example:

  $WSAT: basic

- **$WSCC** - The Client Certificate used for this request. If value is not base64 encoded for us by the webserver then the plugin will base64 encode it before sending it across to the Application Server.

  example:

  $WSCC: CN=jane Doe,O=ibm,C=US

Common Plug-in Request Headers

- **$WSCS** - The Cipher Suite that the webserver negotiated with the client. This is not necessarily the cipher suite that the plugin will use to send the request across to the Application Server.

  example:

  $WSCS: RC4-SHA
- **$WSIS** - This header will be set to either 'true' or 'false' depending on whether or not the request is secure or not(came in over SSL/TLS).

  example:

  $WSIS: true
- **$WSSC** - The scheme being used for the request. This header will normally be set to either 'http' or 'https'.

  example:

  $WSSC: https

Common Plug-in Request Headers

- **$WSPR** - The HTTP protocol level being used for this request. The plugin currently has support for up to HTTP/1.1 requests.
  example:
  $WSPR: HTTP/1.1
- **$WSRA** - The remote IP address of the machine the client is running on.
  example:
  $WSRA: 123.456.789.10
- **$WSRH** - The remote hostname of the machine the client is running on. If the hostname can't be resolved then this header should be set to the IP address.
  example:
  $WSRH: 123.456.789.10

Common Plug-in Request Headers

- **$WSRU** - The remote user specified for the given request.
  example:
  $WSRU: janedoe
- **$WSSN** - The server name used for this request. This should be the value that was specified in the HOST header of the incoming request.
  example:
  $WSSN: www.ibm.com
- **$WSSP** - The server port that the request was received on. This will be the port value that is used in route determination.
  example:
  $WSSP: 443
- **$WSSI** - The SSL Session ID being used for this request. If the value is not base64 encoded for us by the webserver then the plugin will base64 encode it before sending it across to the Application Server.
  example:
  $WSSI: 6CUAAIRUyAQuBXbCq2fjSb/xbQ1YWFhY5ZTTSDoDAAA=

Where Request Headers are logged

- **Packet Traces** (trace.pcap)

  no example provided due to customer confidentiality
- **Web Server access log** (access_log)

  An incoming HTTP request header content can be logged in the IBM HTTP server's access_log using a custom log format, *%{headername}i* and defining this using the LogFormat directive in the httpd.conf.

  example:

  ```
  LogFormat "%v %h %l %u %t \"%r\" %{accept}i %>s %b“
  ```

  example of access_log output:

  ```
  9.37.245.145 - - [25/Sep/2008:15:20:25 -0400] "GET / HTTP/1.1" image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms.powerpoint, application/msword, application/xaml+xml, application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-ms-application, */* 200 3183
  ```

Where Request Headers are logged

- **HTTP plug-in log** (http_plugin.log)

  When loglevel in the plugin-cfg.xml is set to Trace, HTTP request headers which are passed by the HTTP plug-in to WebSphere are logged in the plugin log.

  example:

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

  example of http_plugin.log output:

```
DETAIL:    GET /mystuff/myservlet.jsp HTTP/1.1
DETAIL:    Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/xaml+xml,
application/vnd.ms-xpsdocument, application/x-ms-xbap, application/x-ms-application,
application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, application/x-
silverlight, application/x-shockwave-flash, */*
DETAIL:    Referer: https://www.ibm.com/home.jsp
DETAIL:    Accept-Language: en-ca
DETAIL:    Accept-Encoding: gzip, deflate
DETAIL:    User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR
1.1.4322;.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; InfoPath.1; MS-RTC LM 8)
DETAIL:    Host: www.ibm.com
DETAIL:    Cookie: JSESSIONID=0000CARXl2bhwMkByU9lJkth_ns:12d5ovivu
DETAIL:    $WSCS: DES-CBC3-SHA
DETAIL:    $WSIS: true
DETAIL:    $WSSC: https
DETAIL:    $WSPR: HTTP/1.1
```

etc…

Where Request Headers are logged

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

  example:

```
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   findHeaderLength: Delimiter_found; len=4
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Using bytecache, length=4 pos=48 start=43
[08/08/15 13:17:29:294 JST] 00000041 BinaryHeaderN 3   Byte value [Host] matches key: Key: Host Ordinal: 27
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Found known header: Key: Host Ordinal: 27
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   parseCRLFTokenExtract: start:49 lim:525 pos:49
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   findCRLFTokenLength returning Delimiter_found; len=37
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Saving token: null len:37 start:49 pos:87 delim:true
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Saved token [www.ibm.com]
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Adding known header [Host] with value [www.ibm.com]
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   findHeaderLength: Delimiter_found; len=6
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Using bytecache, length=6 pos=95 start=88
[08/08/15 13:17:29:294 JST] 00000041 BinaryHeaderN 3   Byte value [Accept] matches key: Key: Accept Ordinal: 0
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Found known header: Key: Accept Ordinal: 0
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   parseCRLFTokenExtract: start:96 lim:525 pos:96
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   findCRLFTokenLength returning Delimiter_found; len=67
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Saving token: null len:67 start:96 pos:164 delim:true
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Saved token [image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, image/png, */*]
[08/08/15 13:17:29:294 JST] 00000041 BNFHeadersImp 3   Adding known header [Accept] with value [image/gif,
image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*]
```

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:GenericBNF=all
```

Reference

- For more information on **HTTP Request 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
- 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