Apache Tomcat (Ver 6+)

Tomcat server can be configured to use different type of logging systems, the server has default logging configuration and can be configured to use log4j. Tomcat can also create access logs based on the Access Log Valve.

Tagging

All Tomcat/Catalina logs are tagged by logtype - *tomcat*

In addition there are the following log types that must be assigned for the Tomcat App to be deployed on:

- Catalina out log or Console out log will be tagged by logtype - *out*
- Catalina Servlet or webapp log will be tagged by logtype - *servlet*
- Access logs will be tagged by logtype - *access*

Defualt logging configuration can be found under the conf direcotry (tomcat/conf) or *nix /etc/tomcat../conf/ logging.properties file. Usually the access log will be defined in server.xml file under the conf directory.

Default Log structure and configuration

For log type *out* and *servlet,* those logs can be name by default catalina.out, and will be located under the logs directory. Use the following XpoLog pattern for those logs

Example 1:

{date:Date,dd-MMM-yyyy HH:mm:ss.SSS} {priority:Priority,ftype=severity,ALL;FINEST;FINER;FINE;INFO;CONFIG;WARNING;SEVERE;ERROR} [{text:thread,ftype=thread}] {text:Source,ftype=source} {string:Message,ftype=message}

Custom Logging

If the Tomcat server is configured to use external logging with log4j or other java.util framework than use XpoLog pattern wizard and defenition to configure the log pattern correctly for the app to work.

Make sure that if you are using log4j wizard you will need to setup the log sources and manually apply the Tomcat/Catalina tags on them for the App to work correctly.

References

Tomcat 7

https://tomcat.apache.org/tomcat-7.0-doc/logging.html

Access Logs: https://tomcat.apache.org/tomcat-7.0-doc/config/valve.html#Access_Logging

Tomcat 8

Logging: https://tomcat.apache.org/tomcat-8.0-doc/logging.html

Access Logs: https://tomcat.apache.org/tomcat-8.0-doc/config/valve.html#Access_Logging

Tomcat 9

https://tomcat.apache.org/tomcat-9.0-doc/logging.html

Access Logs: https://tomcat.apache.org/tomcat-9.0-doc/config/valve.html#Access_Logging

Log4J

If the Server is using the log4j library for logging please follow the steps documented in adding logs from log4j 1.2 or log4j 2.*

Tomcat Access Logs Configuration

1. Add Log Data In XpoLog, When adding a log to XpoLog you can now select the Log Type (logtype) for Apache Tomcat Access with the following logtypes:
    a. *tomcat*
        i. in addition select the log type - *access*

Tomcat access logs are created with the *AccessLogValve* or with *ExtendedAccessLogValve* implementation.

For the configuration look into the server <TOMCAT-HOME/conf/server.xml> / (Linux "/etc/tomcat/conf/server.xml") or other webapp configuration files and search for the following:

```
<Engine name="Catalina" defaultHost="localhost">

  <Host name="localhost" ...

     <!-- Access log processes all example.

          Documentation at: /docs/config/valve.html
          Note: The pattern used is equivalent to using pattern="common"
     <Valve className="org.apache.catalina.valves.AccessLogValve"
         directory="logs"
         prefix="localhost_access_log."
         suffix=".txt"
         pattern="%h %l %u %t "%r" %s %b"
      />
   </Host>
</Engine>
```

The ***pattern field*** may defined also as below:

The shorthand pattern `pattern="common"` corresponds to the Common Log Format defined by **%h %l %u %t "%r" %s %b**

The shorthand pattern `pattern="combined"` appends the values of the `Referer` and `User-Agent` headers, each in double quotes, to the `common` pattern.

In XpoLog such pattern (combined) will be translated into:

{text:RemoteHost,ftype=remoteip} {text:logname,ftype=remotelog} {text:Remote User,ftype=remoteuser} [{date:Date,locale=en;,dd/MMM/yyyy:HH:mm:ss z}] "{choice:Method,ftype=reqmethod;,GET;POST;HEAD} {url:URL,paramsFtype=querystring;ftype=requrl;paramsName=Query;,} {string:reqprotocol,ftype=reqprotocol;,}" {number:ResponseStatus,ftype=respstatus} {number:Bytes Sent,ftype=bytesent} "{string:RefererQuery,ftype=refererquery;,}{regexp:Referer,ftype=referer;refName=RefererQuery,^([\w-]+://[^?]+|/[^?]+)}" "{string:User Agent,ftype=useragent;,}"{eoe}

In XpoLog such pattern (common) will be translated into:

{text:RemoteHost,ftype=remoteip} {text:logname,ftype=remotelog} {text:Remote User,ftype=remoteuser} [{date:Date,locale=en;,dd/MMM/yyyy:HH:mm:ss z}] "{choice:Method,ftype=reqmethod;,GET;POST;HEAD} {url:URL,paramsFtype=querystring;ftype=requrl;paramsName=Query;,} {string:reqprotocol,ftype=reqprotocol;,}" {number:ResponseStatus,ftype=respstatus} {number:Bytes Sent,ftype=bytesent}{eoe}

XpoLog Pattern Wizard
When configuring access logs for Tomcat in the XpoLog pattern wizard, paste the pattern directive value into the wizard in order to generate the correct XpoLog pattern for our example you will need to paste: ***%h %l %u %t "%r" %s %b***
Note: If the pattern value is common or combined simply past them into the wizard and XpoLog will build the right pattern as well.

Apache Tomcat Access Log Format Conversion Table both for AccessLogValve and for ExtendedAccessLogValve

logtype should be set to: *tomcat, access*

Format String	Description	XpoLog Pattern
%a	Remote IP-address	{geoip:Remote IP,ftype=remoteip}
%{c}a	Underlying peer IP address and port of the connection	{geoip:Remote IP,ftype=remoteip}
%A	Local IP-address	{ip:Local IP,ftype=localip}
%B	Size of response in bytes, excluding HTTP headers.	{number:Bytes Sent,ftype=bytesent}
%b	Size of response in bytes, excluding HTTP headers. In CLF format, *i.e.* a '-' rather than a 0 when no bytes are sent.	{number:Bytes Sent,ftype=bytesent}

`%{Foobar}C`	The contents of cookie Foobar in the request sent to the server. Only version 0 cookies are fully supported.	{string:Cookie_< FOOBAR >} Replace < FOOBAR > with cookie name
`%D`	The time taken to serve the request, in microseconds.	{number:ResponseTimeMicroSecs,ftype=processrequestmicrosecs}
`%{FOOBAR}e`	The contents of the environment variable FOOBAR	{string:EnvVariable_< FOOBAR >} Replace < FOOBAR > with variable name
`%f`	Filename	{text:FileName}
`%h`	Remote host name (or IP address if `resolveHosts` is false)	{text:Remotehost,ftype=remoteip}
`%H`	The request protocol	{text:RequestProtocol,ftype=reqprotocol}
`%{Foobar}i`	The contents of Foobar`:` header line(s) in the request sent to the server. Changes made by other modules (e.g. mod_headers) affect this. If you're interested in what the request header was prior to when most modules would have modified it, use mod_setenvif to copy the header into an internal environment variable and log that value with the `%{VARNAME}e` described above.	{text:<FOOBAR>} https://en.wikipedia.org/wiki/List_of_HTTP_header_fields and so on it goes for the different headers.
`%{Referer}i`	The referer which is associated with the request	{string:RefererQuery,ftype=refererquery;,}{regexp:Referer,ftype=referer;refName=RefererQuery,^([\w-]+://[^?
`%{User-Agent}i`	The User Agent which is associated with the request	{text:User-agent,ftype=useragent}
`%{`X-Forwarded-For`}i`	Method for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.	{text:X-Forwarded-For,ftype=forwardforip}
`%k`	Number of keepalive requests handled on this connection. Interesting if KeepAlive is being used, so that, for example, a '1' means the first keepalive request after the initial one, '2' the second, etc...; otherwise this is always 0 (indicating the initial request). Available in versions 2.2.11 and later.	{number:KeepAlive}
`%l`	Remote logname (from identd, if supplied). This will return a dash unless mod_ident is present and IdentityCheck is set `On`.	{text:logname,ftype=remotelog}
`%m`	The request method	{choice:Method,ftype=reqmethod;,GET;POST;HEAD}
`%{Foobar}n`	The contents of note Foobar from another module.	{string:<FOOBAR>}

%{Foobar}o	The contents of Foobar: header line(s) in the reply.	{string:<FOOBAR>}
%p	The canonical port of the server serving the request	{number:ServerPort,ftype=serverport}

%{format}p	The canonical port of the server serving the request or the server's actual port or the client's actual port. Valid formats are canonical, local, or remote.
	%{canonical}p
	%{local}p
	%{remote}p
		{number:ServerPort,ftype=serverport}
		{number:LocalServerPort,ftype=localserverport}
		{number:RemotePort,ftype=remoteport}

%P	The process ID of the child that serviced the request.	{text:ProcessID,ftype=processid}
%{format}P	The process ID or thread id of the child that serviced the request. Valid formats are pid, tid, and hextid. hextid requires APR 1.2.0 or higher.	{text:ProcessID,ftype=processid} Valid formats are pid, tid, and hextid.
%{pid}P		{text:ProcessID,ftype=processid}
%{tid}P		{text:ThreadID,ftype=threadid}
%{hextid}P		{text:HexThreadID,ftype=hexthreadid}
%q	The query string (prepended with a ? if a query string exists, otherwise an empty string)	{text:QueryString,ftype=querystring} OR Suggest a regexp that will build a list of parameters as cloumns. The query string (prepended with a ? if a query string exists, otherwise an empty string)
%r	First line of request	1. {choice:Method,ftype=reqmethod;,GET;POST} {url:URL,paramsFtype=querystring;ftype=requrl;paramsName=Query;,} {string:reqprotocol,ftype=reqprc
%R	The handler generating the response (if any).	{text:ResponseHandler}
%s	Status. For requests that got internally redirected, this is the status of the *original* request --- %>s for the last.	{number:ResponseStatus,ftype=respstatus} . For requests that got internally redirected, this is the status of the *original* request --- %>s for the last.
%S	User Session ID	{text:UserSessionID,ftype=sessionid}
%t	Time the request was received (standard english format)	{date:Date,locale=en,dd/MMM/yyyy:HH:mm:ss z}

`%{format}t`	The time, in the form given by format, which should be in an extended `strftime(3)` format (potentially localized). If the format starts with `begin:` (default) the time is taken at the beginning of the request processing. If it starts with `end:` it is the time when the log entry gets written, close to the end of the request processing. In addition to the formats supported by `strftime(3)`, the following format tokens are supported: `sec` number of seconds since the Epoch `msec` number of milliseconds since the Epoch `usec` number of microseconds since the Epoch `msec_frac` millisecond fraction `usec_frac` microsecond fraction These tokens can not be combined with each other or `strftime(3)` formatting in the same format string. You can use multiple `%{format}t` tokens instead. The extended `strftime(3)` tokens are available in 2.2.30 and later.	{date:Date,locale=en,dd/MMM/yyyy:HH:mm:ss z} sec number of seconds since the Epoch msec number of milliseconds since the Epoch usec number of microseconds since the Epoch msec_frac millisecond fraction usec_frac microsecond fraction
`%T`	The time taken to serve the request, in seconds.	{number:ResponseTimeSecs,ftype=processrequestsecs}
`%{UNIT}T`	The time taken to serve the request, in a time unit given by `UNIT`. Valid units are `ms` for milliseconds, `us` for microseconds, and `s` for seconds. Using `s` gives the same result as `%T` without any format; using `us` gives the same result as `%D`. Combining `%T` with a unit is available in 2.2.30 and later.	1. {number:ResponseTimeMilliSecs,ftype=processrequestmilli} 2. {number:ResponseTimeMicroSecs,ftype=processrequestmicrosecs} 3. {number:ResponseTimeSecs,ftype=processrequestsecs}
`%u`	Remote user that was authenticated (from auth; may be bogus if return status (`%s`) is 401)	{string:Remote User,ftype=remoteuser;,} Remote user (from auth; may be bogus if return status (`%s`) is 401)
`%U`	The URL path requested, not including any query string.	{text:RequestURL,ftype=requrl} The URL path requested, not including any query string.
`%v`	The canonical ServerName of the server serving the request.	{text:ServerName,ftype=servername}

`%V`	The server name according to the UseCanonicalName setting.	{text:ServerName,ftype=servername} The server name according to the UseCanonicalName setting.
`%X`	Connection status when response is completed: X = connection aborted before the response completed. + = connection may be kept alive after the response is sent. - = connection will be closed after the response is sent. (This directive was `%c` in late versions of Apache 1.3, but this conflicted with the historical ssl `%{var}c` syntax.)	{text:ConnectionStatus} Connection status when response is completed: X = connection aborted before the response completed. + = connection may be kept alive after the response is sent. - = connection will be closed after the response is sent. (This directive was `%c` in late versions of Apache 1.3, but this conflicted with the historical ssl `%{var}c` synta
`%I`	Bytes received, including request and headers, cannot be zero. You need to enable mod_logio to use this.	{number:TotalBytesWHeadersReceived,ftype=reqbyteswheaders} (with headers)
`%O`	Bytes sent, including headers, cannot be zero. You need to enable mod_logio to use this.	{number:TotalBytesWHeadersSent,ftype=respbyteswheaders} (with headers – can help compute header size)
`%{VARNAME}^ti`	The contents of VARNAME`:` trailer line(s) in the request sent to the server.	{text:Req_*<VARNAME>}* The content of *VARNAME*: trailer line(s) in the request sent to the server.
`%{VARNAME}^to`	The contents of VARNAME`:` trailer line(s) in the response sent from the server.	{text:Resp_*<VARNAME>}* The contents of VARNAME`:` trailer line(s) in the response sent from the server.