

Configuring Connectors

Configuring Connectors

Types of Connectors

org.mortbay.jetty.nio.SelectChannelConnector

This connector uses efficient NIO buffers with a non blocking threading model. Direct NIO buffers are used and threads are only allocated to connections with requests. Synchronization is used to simulate blocking for the servlet API, and any unflushed content at the end of request handling is written asynchronously.

This connector is best used when there are a many connections that have idle periods.

When used with [Continuations](#), threadless waits are supported. When a filter or servlet calls `getEvent` on a Continuation, a runtime exception is thrown to allow the thread to exit the current request handling. Jetty will catch this exception and will not send a response to the client. Instead the thread is released and the Continuation is placed on the timer queue. If the Continuation timeout expires, or it's `resume` method is called, then the request is again allocated a thread and the request is retried. The limitation of this approach is that request content is not available on the retried request, thus if possible it should be read after the continuation or saved as a request attribute or as the associated object of the Continuation instance.

org.mortbay.jetty.bio.SocketConnector

This connector implements a traditional blocking IO and threading model. Normal JRE sockets are used and a thread is allocated per connection. Buffers are managed so that large buffers are only allocated to active connections. This Connector should only be used if NIO is not available.

org.mortbay.jetty.security.SslSelectChannelConnector

SSL connector using NIO. More information about it's configuration can be found on the [Ssl Connector Guide](#) page.

org.mortbay.jetty.security.SslSocketConnector

Blocking IO connector supporting SSL.

org.mortbay.jetty.ajp.Ajp13SocketConnector

Connector implementing the AJP13 protocol. See [Configuring Ajp13](#) for more information, and also take a look at [Configuring mod_proxy](#) for alternatives to using the AJP13 protocol.

Configuration Options

Setter	Description
acceptors	The number of thread dedicated to accepting incoming connections.

acceptQueueSize	Number of connection requests that can be queued up before the operating system starts to send rejections.
acceptorPriority	Sets the priority of the #acceptor threads relative to the other threads.
confidentialPort	The port to redirect to if there is a security constraint of CONFIDENTIAL.
confidentialScheme	https by default
forwarded	If true use #hostHeader or else check the headers for retrieving information from the original request to control what is returned by <code>ServletRequest#getServerName()</code> and <code>ServletRequest#getServerPort()</code> and <code>ServletRequest#getRemoteAddr()</code> (see Configuring mod_proxy). Default is false.
forwardedHostHeader	The forwarded Host header to use. Default is X-Forwarded-Host. This value is only used if #forwarded is true.
forwardedServerHeader	The forwarded server name header to use. Default is X-Forwarded-Server. This value is only used if #forwarded is true.
forwardedForHeader	The forwarded for header to use. Default is X-Forwarded-For. This value is only used if #forwarded is true.
headerBufferSize	Set the size of the buffer to be used for request and response headers. An idle connection will at most have one buffer of this size allocated. Default is 4K.
hostHeader	Set a forced valued for the Host header to control what is returned by <code>ServletRequest#getServerName()</code> and <code>ServletRequest#getServerPort()</code> . This value is only used if #forwarded is true.
host	The particular interface to listen on. If not set or 0.0.0.0, jetty will listen on #port on all interfaces.
integralPort	The port to redirect to if there is a security constraint of INTEGRAL.

lowResourcesConnections	Set the number of connections, which if exceeded places this connector in a low resources state. This is not an exact measure as the connection count is averaged over the select sets. When in a low resources state, different idle timeouts can apply on connections (see #lowResourcesMaxIdleTime).
lowResourcesMaxIdleTime	Set the period in ms that a connection is allowed to be idle when there are more than #lowResourcesConnections connections. This allows the server to rapidly close idle connections in order to gracefully handle high load situations.
maxIdleTime	Set the maximum idle time for a connection, which roughly translates to the <code>Socket.setSoTimeout(int)</code> call, although with NIO implementations other mechanisms may be used to implement the timeout. The max idle time is applied: when waiting for a new request to be received on a connection; when reading the headers and content of a request; when writing the headers and content of a response. Jetty interprets this value as the maximum time between some progress being made on the connection. So if a single byte is read or written, then the timeout (if implemented by jetty) is reset. However, in many instances, the reading/writing is delegated to the JVM, and the semantic is more strictly enforced as the maximum time a single read/write operation can take. Note, that as Jetty supports writes of memory mapped file buffers, then a write may take many 10s of seconds for large content written to a slow device.
name	The name of the connector. Can be used to make a <code>WebAppContext</code> respond only to requests on the named connector via the WebAppContext.setConnectorNames(String[]) method
port	The port to listen on. See also #host
requestBufferSize	Set the size of the content buffer for receiving requests. These buffers are only used for active connections that have requests with bodies that will not fit within the header buffer (see #headerBufferSize). Default is 8K.
responseBufferSize	Set the size of the content buffer for sending responses. These buffers are only used for active connections that are sending responses with bodies that will not fit within the header buffer. Default is 32K.

resolveNames	If true, request IP addresses will be resolved to host names
reuseAddress	True if the the server socket will be opened in SO_REUSEADDR mode
	Sets SO_LINGER on the connection socket. Disabled by default.
statsOn	If true, enables statistics collection on connections see Statistics
useDirectBuffers	For nio connectors, determines whether direct byte buffers will be used or not. The default is true.
threadPool	Sets the thread pool instance. By default this is the thread pool set on the org.mortbay.jetty.Server , and is a org.mortbay.thread.QueuedThreadPool instance.

Configuration Differences for Jetty 6

Setter	Description
delayedKeyUpdate	For non-blocking channels, whether or not the selection key is left in the select set whilst handling the io. If left in the select set it can be an optimization because frequently there will be more io to do. Default is true.
responseBufferSize	Set the size of the content buffer for sending responses. These buffers are only used for active connections that are sending responses with bodies that will not fit within the header buffer. Default is 24K.