

Introduction to Jetty6

Introduction

Jetty 6 is a fully featured web server for static and dynamic content. Unlike separate server/container solutions, this means that your web server and web application run in the same process, without interconnection overheads and complications. In addition, as a pure Java component, Jetty 6 can be simply included in your application for demonstration, distribution or deployment. It is available on all Java supported platforms. Open Source Jetty 6 is developed under the guidance of Mort Bay Consulting.

Jetty 6 is the new implementation of Jetty HTTP Server and Servlet Container. Unlike the older version of Jetty, Jetty 6 focuses on a 3rd party dependencies by reducing its previous dependencies so that it would be easier to embed it with other applications. Jetty 6 has been developed with the concrete foundation of technology which supports the different kinds of open source technologies (**AJAX Continuations, NIO, JAVA API's**).

Jetty 6 comes with the new set of connectors based on Sun's non blocking IO (NIO) which can be found in `org.mortbay.jetty.nio` package. The NIO libraries can help, as it allows asynchronous IO to be used and threads can be allocated to connections only when requests are being processed. When the connection is idle between requests, then the thread can be returned to a thread pool and the connection can be added to an NIO select set to detect new requests. This thread-per-request model allows much greater scaling of connections (users) at the expense of a reduced maximum requests per second for the server as a whole (in Jetty 6 this expense has been significantly reduced).

Jetty 6 supports a new feature called Continuations that will allow scalable AJAX applications to be built, with threadless waiting for asynchronous events. As per scalability it is one of the major concerns of this product that would be supported by the advent of AJAX with the response to offer a feature called Continuations. Where in it allows a request to be suspended and resumed later in the process. In addition it is a Java Filter or Servlet that is handling an AJAX request, may now request a Continuation object that can be used to effectively suspend the request and free the current thread. The request is resumed after a timeout or immediately if the resume method is called on the Continuation object. Jetty 6 employs a number of innovative strategies to ensure that only the resources that are actually required are assigned to a connection and only for the duration of they are needed. This careful resource management gives Jetty an architecture designed to scale to meet the needs of AJAX applications

The summary features of Jetty6:

- Codebase completely rearchitected and slimlined
- Continuations API for efficient AJAX polling
- Improved dependency injection and inversion of control design of components
- Improved interceptor design of handlers
- Smart split buffer design for efficient asynchronous flushing and avoiding chunking.
- Optional use of NIO Buffering so that efficient direct buffers and memory mapped files can be used.
- Optional use of NIO non-blocking scheduling so that threads are not allocated per connection.
- Optional use of NIO gather writes, so that for example a HTTP header and a memory mapped file may be sent as a single operation.