Atitit.切入式web server jetty

# 为什么使用jetty

方便基于hre程序的调试与开发。。只要一个web版即可。。

# 下面看一下 Jetty 是如何处理一个 HTTP 请求的。

实际上 Jetty 的工作方式非常简单，当 Jetty 接受到一个请求时，Jetty 就把这个请求交给在 Server 中注册的代理 Handler 去执行，如何执行你注册的 Handler，同样由你去规定，Jetty 要做的就是调用你注册的第一个 Handler 的 handle(String target, Request baseRequest, HttpServletRequest request, HttpServletResponse response) 方法，接下去要怎么做，完全由你决定。

要能接受一个 web 请求访问，首先要创建一个 ContextHandler，如下代码所示：

|  |
| --- |
| Server server = new Server(8080);  ContextHandler context = new ContextHandler();  context.setContextPath("/");  context.setResourceBase(".");  context.setClassLoader(Thread.currentThread().getContextClassLoader());  server.setHandler(context);  context.setHandler(new HelloHandler());  server.start();  server.join(); |

当我们在浏览器里敲入 http://localhost:8080 时的请求将会代理到 Server 类的 handle 方法，Server 的 handle 的方法将请求代理给 ContextHandler 的 handle 方法，ContextHandler 又调用 HelloHandler 的 handle 方法。这个调用方式是不是和 Servlet 的工作方式类似，在启动之前初始化，然后创建对象后调用 Servlet 的 service 方法。在 Servlet 的 API 中我通常也只实现它的一个包装好的类，在 Jetty 中也是如此，虽然 ContextHandler 也只是一个 Handler，但是这个 Handler 通常是由 Jetty 帮你实现了，我们一般只要实现一些我们具体要做的业务逻辑有关的 Handler 就好了，而一些流程性的或某些规范的 Handler，我们直接用就好了，如下面的关于 Jetty 支持 Servlet 的规范的 Handler 就有多种实现，下面是一个简单的 HTTP 请求的流程。

访问一个 Servlet 的代码：

|  |
| --- |
| Server server = new Server();  Connector connector = new SelectChannelConnector();  connector.setPort(8080);  server.setConnectors(new Connector[]{ connector });  ServletContextHandler root = new  ServletContextHandler(null,"/",ServletContextHandler.SESSIONS);  server.setHandler(root);  root.addServlet(new ServletHolder(new  org.eclipse.jetty.embedded.HelloServlet("Hello")),"/");  server.start();  server.join(); |

创建一个 ServletContextHandler 并给这个 Handler 添加一个 Servlet，这里的 ServletHolder 是 Servlet 的一个装饰类，它十分类似于 Tomcat 中的 StandardWrapper。下面是请求这个 Servlet 的时序图：