

Jetty6 testing

How to test using an embedded instance of Jetty6

This is a note to describe how to embed an instance of Jetty6 in order to test in the same way as the QuickStart project currently does using Jetty4.

The basic change involves replacing the code in the 'Start' class that passes the jetty-config.xml to the Jetty4 server with Java code that does the minimal setup required for Jetty 6, shown below...

With this change, the jetty-config.xml is no longer used - There may be a way of still passing a config file, but I'm not aware of it at the moment.

```
jettyServer = new Server();

SocketConnector conn = new SocketConnector();
conn.setPort(8080);
jettyServer.setConnectors(new Connector[]{conn});

WebAppContext wah = new WebAppContext();
wah.setContextPath("/myapp");
wah.setWar("src/main/webapp");
jettyServer.setHandler(wah);

jettyServer.start();
```

Just for completeness, that results in a 'Start' class as follows.

(Feel free to replace the logger with your own preferred one, of course, or see <https://simple-log.dev.java.net/> for more info on that one!)

```

package com.smarttrust.cd.mysp.tpm;

import org.grlea.log.SimpleLogger;
import org.mortbay.jetty.Connector;
import org.mortbay.jetty.Server;
import org.mortbay.jetty.bio.SocketConnector;
import org.mortbay.jetty.webapp.WebAppContext;

public class Start {
    private static final SimpleLogger log = new SimpleLogger(Start.class);

    /**
     * Main function, starts the Jetty 6 server.
     *
     * @param args
     */
    public static void main(String[] args) {
        Server jettyServer = null;
        try {
            jettyServer = new Server();

            SocketConnector conn = new SocketConnector();
            conn.setPort(8080);
            jettyServer.setConnectors(new Connector[]{conn});

            WebAppContext wah = new WebAppContext();
            wah.setContextPath("/myapp");
            wah.setWar("src/main/webapp");
            jettyServer.setHandler(wah);

            jettyServer.start();
        }
        catch (Exception e) {
            log.fatal("Could not start the Jetty server: " + e);
            if (jettyServer != null) {
                try {
                    jettyServer.stop();
                }
                catch (Exception e1) {
                    log.fatal("Unable to stop the jetty server: " + e1);
                }
            }
        }
    }
}

```

SSL version of the above

A keystore is needed (can be generated with \$JAVA_HOME/bin/keytool -genkey -keystore keystore).

```

import org.mortbay.jetty.Connector;
import org.mortbay.jetty.Server;
import org.mortbay.jetty.security.SslSocketConnector;
import org.mortbay.jetty.webapp.WebAppContext;

public class StartSsl {

    public static void main(String[] args) {
        String password = System.getProperty("keystorepassword", "jetty6");
        String keystore = System.getProperty("keystore", "src/main/resources/keystore");
        String war = System.getProperty("warpath", "src/main/webapp");
        String contextPath = System.getProperty("contextpath", "/");
        int sslPort = Integer.parseInt(System.getProperty("port", "8443"));

        Server jettyServer = null;
        try {
            jettyServer = new Server();

            SslSocketConnector sslConn = new SslSocketConnector();
            sslConn.setPort(sslPort);
            sslConn.setPassword(password);
            sslConn.setKeystore(keystore);

            jettyServer.setConnectors(new Connector[] { sslConn });

            WebAppContext wah = new WebAppContext();
            wah.setContextPath(contextPath);
            wah.setWar(war);
            jettyServer.setHandler(wah);

            jettyServer.start();
        } catch (Exception e) {
            System.out.println("Could not start the Jetty server: " + e);
            if (jettyServer != null) {
                try {
                    jettyServer.stop();
                } catch (Exception e1) {
                    System.out.println("Unable to stop the jetty server: " + e1);
                }
            }
        }
    }
}

```