

StaxMateSampleWebService

Sample Web Service for StaxMate

To give an idea of relative simplicity of using StaxMate, I wrote a simple web service using StaxMate. And to avoid writing a trivial and useless piece of sample code, I decided to embed actual piece of functionality that someone might want to use (even if not as a web service?): Universal Unique Identifier, UUID (also known as GUIDs in Windows world) generation functionality .

Implementation Overview

Web service interface is of "REST" (or, Plain Old Xml over HTTP, aka POX/HTTP) variety, so that sample can focus on aspects of StaxMate instead of the transport protocol. Both GET and POST methods can be used; with GET, only query parameters can be used, with POST, input needs to be in xml, and query parameters will be ignored.

The service has only one method, used to generate UUIDs (also known as GUIDs in Windows world).

Web service is built as a standard web application, deployed as a .war file (on any servlet container, although [Jetty 6](#) is bundled with the sample archive, and warmly recommended).

The actual UUID generation is done using [Java Uuid Generator](#) , but xml processing (request parameter parsing if passed via POST, output writing) is done using StaxMate.

Where can I get it?

For the impatient, you can find download:

- [Source archive](#)
- [Server .war](#)
- To be added: ~~Client.jar~~

Of which you will usually need just the first one, which can build the other two.

How can I use it?

There are multiple ways. One possibility is to download the full source archive (either one mentioned above; or full source code of StaxMate itself, sample web service can be found from under **PROJECT-ROOT/sample-webapp**), and:

- Build and deploy web service: `*ant build deploy`
- Start bundled Jetty server with the built web app: `./run-webapp.sh` (note: may need to change invocation for non-unix systems)

Alternatively, you can just download the web archive (war) file mentioned above, and deploy it under your favourite app server / servlet container.

In either case, there are multiple ways you can access the service:

- Using direct URL to invoke generation method, from your browser: `*http://localhost:7272/uuid-server/generate-uuid?method=RANDOM*` (note: port may be different if you change start script)
- Using simple Ajax-enhanced test web page included as part of the web app: point your browser to address like `*http://localhost:7272/uuid-server/*`
- (TODO) Use the included simple web-service client from your code

Sample classes

There is only one java class, the main servlet that implements the service:

- [UuidServlet.java](#)

The main things to note about the Servlet (wrt StaxMate) are:

- Method **doPost** (lines 97 - 144) handles incoming xml requests.
- Method **writeResponse** (lines 146 - 165) handles writing of success response
- Method **reportProblem** (lines 180 - 214) handles writing of error response

(to be continued)