Asynchronous REST

Asynchronous REST example

This demonstration shows how the proposed servlet 3.0 asynchronous API (inspired by Jetty continuations) can be used to greatly improve web services. The demo runs searches using the eBay API, both synchronously and asynchronously, and displays the results side-by-side.

The synchronous servlet calls the web service for each keyword sequentially. The thread servicing the web request blocks, waiting for each search to complete before making the next request. The servlet thread is held for the entire elapsed time of all web service calls.

The asynchronous servlet uses the Jetty HTTP Client to initiate an asynchronous web service call for all keywords in parallel. As each search completes, its results are added to the list of search results using a callback method. After all searches have been completed, the request is resumed. The web service calls are done in parallel and the servlet thread is only held for a short time at the start and end of the elapsed time.

With one item, the time it takes to fetch the search results is roughly equivalent for both synchronous and asynchronous calls. However, with each additional items, the time it takes to search using the synchronous servlet increases proportionally, while the asynchronous servlet incurs only minor additional overhead.

Usage

To run the demonstration, first get the source code from SVN. Run it using

```
mvn jetty:run
```

then visit http://localhost:8080/. There are two examples each of synchronous and asynchronous; the first example searches for one item, the second for multiple.

You can also use the form at the bottom of the page to run your own searches. For more dramatic results, try conducting a search with ten different items at once.