

Web Services Support

by Don Brown

Advanced configuration

Table of contents

1 Overview.....	2
2 REST-style Web Services.....	2

1. Overview

In addition to exposing an HTML interface for humans to interact with the data a stxx application provides, it also might be desirable to take those same actions and expose their information as pure data without any formatting for other applications to use. The exposing of pure XML data over the web is loosely known as "web services".

2. REST-style Web Services

While there are several different architectural styles for constructing web services, stxx supports a style known as [REST](#). Other good sources of information are [here](#) and [here](#). Taken from the latter, here is a short unofficial list of what are REST best practices:

1. Every public resource should have a URI.
2. A URI should expose no detail of its implementation; in Costello's terms, they should be "logical" rather than "physical".
3. Since resources are objects rather than processes, URI parts should be named for nouns rather than verbs.
4. Responses to HTTP GET should be free of side effects.
5. The representation of a resource should contain links to other resources.
6. Eliminate "query strings" as often as possible.

Unfortunately, Struts is very much verb oriented (violating number 3) and requires each action mapping URL to be explicitly named (violating number 5). To better support REST-style web services, stxx incorporated code from the [Wildcard-Matched Actions](#) project to allow action mappings to use wildcards.

To demonstrate how wildcard-matched actions work, say, for example, a web application needed to display an item. Struts would require the action mapping to look something like this:

```
<action path="/viewItem"
        type="foo.ViewItemAction">
    <forward name="success" path="xml/view-item.dox" />
</action>
```

With wildcards:

```
<action path="/item/*"
        type="foo.ViewItemAction">
    parameter="{1}"
    <forward name="success" path="xml/view-item.dox" />
</action>
```

With the former, to request an item, the URL would look like this:
`http://server/webapp/viewItem.do?key=1`. With wildcards, it would look like this: `http://server/webapp/item/1.do`. While it isn't perfect (the ".do" suffix isn't desirable), it is much closer to REST-style best practices.