

Web Scripting for Java

Asked 16 years, 3 months ago Modified 16 years, 3 months ago

Viewed 427 times



What is a good way to render data produced by a Java process in the browser?

0



I've made extensive use of JSP and the various associated frameworks ([JSTL](#), [Struts](#), [Tapestry](#), etc), as well as more comprehensive frameworks not related to JSP ([GWT](#), [OpenLaszlo](#)). None of the solutions have ever been entirely satisfactory - in most cases the framework is too constrained or too complex for my needs, while others would require extensive refactoring of existing code. Additionally, most frameworks seem to have performance problems.



Currently I'm leaning towards the solution of exposing my java data via a simple servlet that returns JSON, and then rendering the data using PHP or Ruby. This has the added benefit of instantly exposing my service as a web service as well, but I'm wondering if I'm reinventing the wheel here.

java

jsp

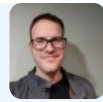
scripting

asked Sep 17, 2008 at 14:59

Share

Improve this question

Follow



toluju

4,117 ● 2 ● 25 ● 27

4 Answers

Sorted by:

Highest score (default)



2



I personally use [Tapestry 5](#) for creating webpages with Java, but I agree that it can sometimes be a bit overkill. I would look into using JAX-RS ([java.net project](#), [jsr311](#)) it is pretty simple to use, it supports marshalling and unmarshalling objects to/from XML out of the box. It is possible to extend it to support JSON via [Jettison](#).

There are two implementations that I have tried:

- [Jersey](#) - the reference implementation for JAX-RS.
- [Resteasy](#) - the implementation I prefer, good support for marshalling and unmarshalling a wide-range of formats. Also pretty stable and has more features than Jersey.

Take a look at the following code to get a feeling for what JAX-RS can do for you:

```
@Path("/")
class TestClass {
    @GET
    @Path("text")
    @Produces("text/plain")
    String getText() {
        return "String value";
    }
}
```

```
}  
}
```

This tiny class will expose itself at the root of the server (@Path on the class), then expose the getText() method at the URI /text and allow access to it via HTTP GET. The @Produces annotation tells the JAX-RS framework to attempt to turn the result of the method into plain text.

The easiest way to learn about what is possible with JAX-RS is to read the [specification](#).

Share Improve this answer

answered Sep 17, 2008 at 20:11

Follow



[Andreas Holstenson](#)

1,428 ● 9 ● 7



1



We're using [Stripes](#). It gives you more structure than straight servlets, but it lets you control your urls through a @UrlBinding annotation. We use it to stream xml and json back to the browser for ajax stuff.



You could easily consume it with another technology if you wanted to go that route, but you may actually enjoy developing with stripes.

Share Improve this answer

answered Sep 17, 2008 at 15:04

Follow



[ScArcher2](#)

87.2k ● 47 ● 122 ● 160



1

Check out [Restlet](#) for a good framework for exposing your domain model as REST services (including JSON and trivial XML output).



For rendering your info, maybe you can use GWT on the client side and consume your data services? If GWT doesn't float your boat, then maybe JQuery would?



Share Improve this answer

answered Sep 17, 2008 at 15:04

Follow



[Tim Howland](#)

7,970 ● 4 ● 29 ● 46



0

Perhaps you could generate the data as XML and render it using XSLT?

I'm not sure PHP or Ruby are the answer if Java isn't fast enough for you!



Share Improve this answer

answered Sep 17, 2008 at 15:04

Follow



[user7094](#)