

Wednesday, February 21, 2007

OpenID library for the Zend Framework?


A few months ago I was requested to add OpenID support (a growing trend) to a small website. Some of the members of the society whose website this is, were running blogs and Flickr sites and thought it would be "neat". Being a fan of "neat" myself, I figured it would be reasonable to add, and indeed it was. I was using the JanRain PHP-Openid library seeing as it is likely the only full implementation in PHP so far.

Here in the present, everyone I work with is all too aware of OpenID. It's been growing up, especially with the current 1.2 set of proposals, and of course support for OpenID is bound to be spurred higher by recent support announcements by [Mozilla for Firefox 3](#), and [Microsoft for Windows Vista](#). Also, the open Yadis specification is useful outside of OpenID for other things.


Loving PHP5 and being a long time OOP fanboy, I've decided to take on a small personal project - writing a strictly PHP5 library to support Yadis and OpenID respectively. Partly to learn the teeny details surrounding OpenID/Yadis, and also to add a pure PHP5 library which is strictly object oriented. Of course such an undertaking is a complete pain in the ass since OpenID requires a lot of functionality from cryptographic algorithms (HMAC/Diffie Hellman), big integer math (bcmath/gmp/big_int support) to simple things like making HTTP requests from PHP.


Enter the Zend Framework. The ZF has become central to no few of my projects. I also use other frameworks, but I just find the ZF hits the sweet spot a lot. Since the Zend Framework has a lot of independent components, relying on it lets me take advantage of components like Zend_Service_Abstract, Zend_Uri and Zend_Http_Client among others. If you've ever looked under the hood of the JanRain PHP-Openid library you'll notice they implement **all** of this without a single dependency on PEAR or other external classes. I don't want to repeat that practice here.

I guess I can be accused of "Not Invented Here" syndrome. I'm willing to plead guilty on that. In my defence, there exists only the one (AFAIK) php library dedicated to OpenID. It's written in PHP4 and mixes procedural functions into its otherwise OOP approach. I found it unnecessarily complex and duplicative - a lot of its functionality is readily available from PEAR or other sources. But outside my personal impressions, I love tinkering with and learning the nitty-gritty of new ideas. I do hope the library will prove acceptable enough to eventually be added to the Zend Framework. Regardless of that suggestion I intend plundering the ZF for all its worth, and release everything under the New BSD license anyway.

In the meantime, as a bit of a brain dump exercise I'll be posting a lot more about the technologies underlying OpenID. It's a well documented area, but I doubt everyone reading this blog has the time to read through all the associated specifications and their recent proposal updates. Should be a fun journey - for me anyway 

Posted by Pádraic Brady in PHP General, PHP Security, Zend Framework at 06:48

Bravo! I implemented a login using JanRain's OpenID package back in January and found it definitely lacking. As soon as I found the time, I had every intention of writing a new PHP5 one that was coded a little more cleanly and actually got installed in the correct place. 

Hopefully, you'll get something done and I won't have to do that though... 

If you need any help, just shoot me an email. Anonymous

on Feb 21 2007, 07:43

hi your site very good thanks webmaster. good bye Anonymous on Feb 25 2007, 04:02

How is the Zend OpenID / new OO plug-in going?

Have you set the project up anywhere, it would be good to work on it as a community? Anonymous on Mar 3 2007, 00:23


I set up a subversion repository to start work on the Yadis component, and so a little experimenting.
<http://svn.astrumfutura.org/zendframework>

The Yadis files are located at:
<http://svn.astrumfutura.org/zendframework/trunk/library/Zend/Service/>

Under the Zend_Service prefix. I'd say it's about 85-90% done for Yadis. A few bits and pieces like Priority ordering is not yet added but small additions like that are easily added.

If you want to help out - drop me an email to maugrimtr AT hotmail DOT com. I can then setup a small mailing list if anyone else interested drops by. If you're seriously considering contributing (always welcome!) you might consider checking out the "Contributor License Agreement" for the Zend Framework. It's basically an agreement to be sent to Zend (Bill or Gavin I think) which just verifies you will not contribute code from other licensed (non-BSD) sources you don't own, and you won't sue anyone for patent infringement. Simple three pager you can sign and email back.

This will get you access to the Framework Wiki and Tracker (JIRA) as a user, and also qualify you for later svn access once the Yadis/OpenID proposal is reviewed.

Long reply 

Anonymous on Mar 6 2007, 00:47

Quick note - you should checkout the entire trunk from subversion since it contains all the Zend Framework classes the Yadis component relies on. After that you can run a few quick tests to try it out. I haven't tested it exhaustively, for example XRI/i-name support is not finished. But any URI based ID should work. It's well commented, but if you need a specific example of it's usage I can post a few example in a later blog entry. Anonymous on Mar 6 2007, 00:55