

Friday, March 4, 2005

PHP5-Client for Yahoo's Webservice API

Our new member Daniel Wiegand (wiegi) implemented a PEAR compatible client for Yahoo's new REST-based webservice API. From the Yahoo developer site: "Yahoo! Search Web Services allow you to access Yahoo content and services in your favorite programming languages. This means you can now build Yahoo directly into your own applications." By using Daniel's PHP5 client you do not need to worry about the protocol or the resulting XML format, everything is done in plain PHP:

Besides `searchWeb()` the client provides more methods to use the different web services offered by Yahoo: `searchImages()`, `searchVideos()`, `searchNews`, `searchLocal()`. Each of the methods can be used with several parameters that need to be passed in a fixed parameter order or you may pass an associative array that contains any parameters you want to pass.

As Daniel has been using PHP5 all of these methods return a `Services_Yahoo_Result` object that provides methods to return the total number of search results but can be iterated like a normal object, thanks to SPL. This result object offers a lot more, as you can use it to fetch the next page using the same search parameters:

Quite cool, isn't it?

You can take a look at the sources at http://pear.php-tools.net/projects/Services_Yahoo/ or even download a PEAR installable package from http://pear.php-tools.net/projects/Services_Yahoo/Services_Yahoo-0.1.0.tgz (requires PHP5, XML_Serializer and HTTP_Request).

We would have loved to propose this to PEAR, but there already is a draft for a `Services_Yahoo` package by Martin Jansen, but until now he hasn't showed any of his code. Hopefully Daniel and Martin could be working on a package together.

Posted by schst in PHP at 12:05

FWIW, the source code of my draft is now available at http://pear.php.net/~mj/Services_Yahoo/.

Usage examples will follow.

Anonymous on Mar 4 2005, 14:21

Looks good as well. I think both could be easily merged to one class combining all the advantages:

- Although your class is split nicely in lot of classes for separate tasks using it can get quite verbose. Having some wrappers for this would be nice
- Being able to call a `fetchNext()` on the result object as Daniel's implementation is able to would be cool as well.

Just my 2 cts.

Stephan

Anonymous on Mar 4 2005, 14:32