

The wpSearch Manual

Yes, a manual. 2008.07.20

- Basic Installation
- Usage
- Playing with wpSearch Options
- Extra functionality
- Comments

Basic Installation

wpSearch requires PHP version 5. This is due to the fact that the internal search library, Zend Lucene, requires PHP 5. wpSearch also requires Wordpress version greater than 2.2.x.

Steps 1 – 3, and 5 are the generic steps for installing any search plugin

Steps:

- 1. Download wpSearch:** Just in case you haven't yet, download wpSearch to your web server.
- 2. Unzip / Decompress Files:** Unzip the file you downloaded. You should now have a single directory named 'wpSearch' with a number of files and directories inside.
- 3. Move wpSearch:** Move wpSearch into your Wordpress plugins directory. This is at: /WP-content/plugins/
- 4. Make sure permissions are set correctly:** This step is only necessary on some machines. You may skip this option now if you would like. If there is an error in step 6, complete this step and try step 6 again.

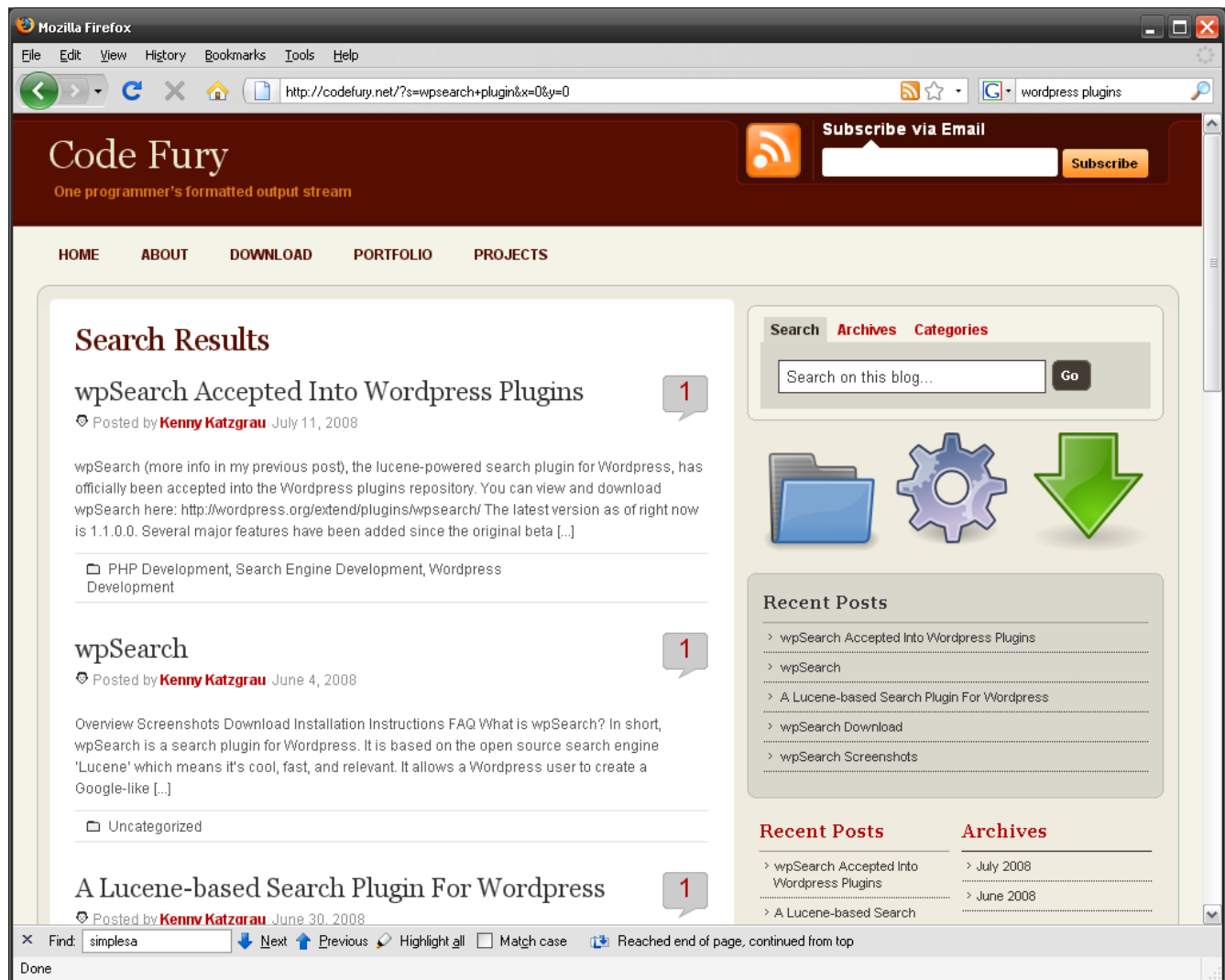
If you are running Wordpress on a Windows machine, make sure that the Wordpress directory and its subdirectories are **not** read-only. If you are running Wordpress on a Unix-like machine, make sure that the wpSearch directory has full read, write, and execution permissions. This can be done with “chmod 777 wpSearch” in the shell within the plugins folder. You can also do this via your FTP client. [This Wordpress tutorial goes a little more in-depth.](#)

- 5. Activate wpSearch:** Log into the Wordpress administration page, and click “Plugins” on the right side of the screen. On this page, wpSearch will be listed in the list of plugins. Find the entry for wpSearch, and click the “Activate” link. A message stating that the plugin was activated should appear after the page reloads. If you receive an error, there is a bug in wpSearch, and you deserve to send a mean email to katzgrau@gmail.com.
- 6. Build your search index:** On the right side of the Wordpress administration page, click “Settings.” On the sub-menu which appears, click the “wpSearch Options” link. On the options page, check the box that says “Build Search Index” next to it. Click “Save changes.” If all goes well, the page will reload say “Building Index ... Complete”. If you receive an error, there is a bug in wpSearch, and you deserve to send a mean email to katzgrau@gmail.com.

Usage

After installation, the search form on your blog will be configured to use wpSearch. If for some reason this is not the case, let me know at katzgrau@gmail.com.

When you now submit a search on your blog, your results will be loaded onto the page. The results' styling should look exactly as they did before installing the plugin. The only difference you will notice is that your searches are faster, and more relevant. This is what the search results look like on my blog:



If you have a blog theme that doesn't actually contain a search form (which I actually haven't encountered, but I assume it's possible), you can still implement wpSearch on your blog with the instruction provided in the following section. You can also switch to a theme that includes a search box.

Creating a “Search Page” if your blog does not have a search form (or you simply want one)

In this method we will create a page on your Wordpress blog which does not contain a search box. This method is very basic and requires very little technical knowledge. An example of the implementation is running at: <http://codefury.net/search-codefury/>

Steps:

- 1. Create a New Page:** Create a new page on your blog, with any title you would like (an example would be “Search This Blog”). When editing the page, click the “HTML” link on the top right corner of the editor to edit the HTML. Copy and paste the following into the editor:

```
Hi! You can search my blog! <br />

<form method="get" id="searchform" action="<?php
bloginfo('home'); ?>">
<input type="text" value="" name="s" />
<input type="button" value="Search" />
</form>
```

After saving and publishing the page, you will be able to view the page you just created, and use the search form.

If this process did not work for you, feel free to send an email to katzgrau@gmail.com.

Playing With wpSearch Options

There are four things on the settings page (mentioned in step 6 in the install section) that a user can do:

- Tweak the relevancy settings of a post
- Rebuild the blog search index

Tweak Relevancy: Every post has a title, content, and sometimes tags. These fields can be 'boosted', which means that you can make one field more important than another, to any degree.

For example, if the Title field's boost is 2, and the content and tag field's boost is 1, I am essentially saying that searches matching the title of a post are twice as relevant as those matching the content of a post only.

These numbers usually range for 0.0 or 1.0 to 2.0. They can be numbers with decimals. The numbers supplied by wpSearch's default setup will be perfect for most bloggers.

(Re)Build the search index: You shouldn't have to do this, but if for some reason the search index is corrupted, or search result links seem to be incorrect, it may be because the search index has been corrupted. Rebuilding the index will fix this.

Extra Functionality

Once the index is built, searches can be made via POST or GET to `wpSearchService.php`. A call to `http://blogURL/wp-content/plugins/wpSearch/wpSearchService.php` will return results under certain parameters. All results are returned in JSON format.

Parameter Listing

Parameter	GET or POST ?	Description
q	Both	The content of the search query. This should be escaped for safe URL searching.

p	Both	The page number you would like returned from the search service.
n	Both	The number of results per page (regarding parameter p).

Example:

Suppose I wanted to create my own little hack with wpSearch. If I wanted to search for "lucene" on my blog, with 5 results per page, on the second page, I would call:

```
http://[MY-BLOG-URL]/wp-content/plugins/wpSearch/wpSearchService.php?  
q=lucene&n=5&p=2
```

An example at code fury would be:

```
http://codefury.net/wp-content/plugins/wpSearch/wpSearchService.php?  
q=lucene&n=1&p=1
```

Comments

Just a few notes:

- wpSearch uses components of the free Zend Framework at: <http://framework.zend.com/>
- wpSearch uses a StandardAnalyzer developed by myself, which uses an algorithm written by Richard Heyes at <http://phpguru.net>
- wpSearch was written using Notepad++, far and away the greatest Windows notepad alternative. Check it out at: <http://notepad-plus.sourceforge.net/uk/site.htm>
- wpSearch is no longer Javascript dependent, and no longer requires any CSS or Javascript includes on your blog. All search routines are integrated into Wordpress.

Once again, send comments and bug reports to katzgrau@njit.edu