

Create Your Own Search Engine with PHP and Google Web Services

Author: Ahm Asaduzzaman
Date Added: 12th Feb 2003
Type: Tutorial
Rating: 

This is a printable version of "Create Your Own Search Engine with PHP and Google Web Services". For the complete online version, please visit <http://www.devarticles.com/content.php?articleId=422>

Page 1: Introduction

The buzz about Web Services gets louder every day. Is it the promise of perfect interoperability, lower costs, and increased efficiency? In this article, an effort has taken to show you how to create your own "Google search engine" with [Web Services](#) provided by Google. First, you need to create a Google account [here](#) (all you need is an email address).

In order to invoke the Web Services we also need the toolkit. In this article we will use PHP NuSoap classes; free download [here](#). I strongly recommend seeing the [FAQ](#) page to know all about Google's cutting edge technology available to the public. If you are interested in the advanced theory of the search engine, see [this publication](#) from Stanford University.

Page 2: Interacting with Google's Web Services

Google has made only three methods available in their Web API. You will see soon that it's just more than enough to build powerful applications.

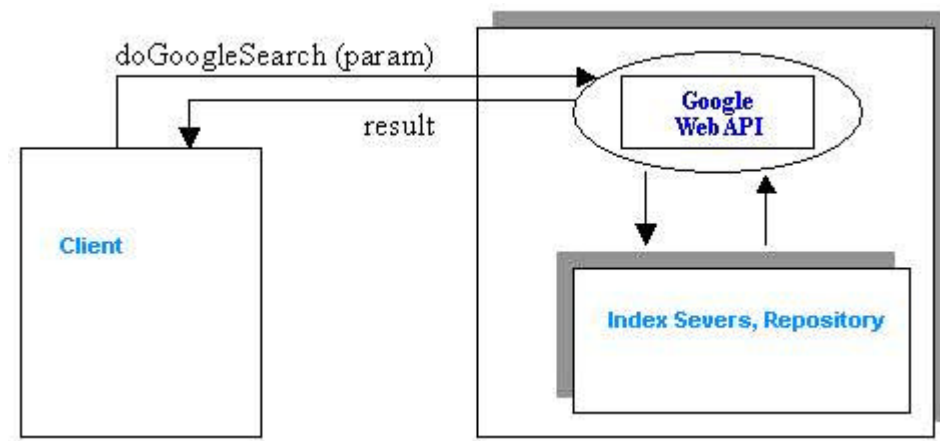
Here is what they look like:

`doGoogleSearch()` – search for specified term in the Google database.

`doGetCachedPage()` – retrieve a page cache from the Google cache.

`doSpellingSuggestion()` – retrieve a spelling suggestion from Google.

The above methods exposed by the Google's Web Services make it possible to use number of very cool features – Web search, cached document retrieval, phrase correction – in a simple but extensible manner, which opens up very interesting new possibilities for Web developers. Pic. 1 shows how a client program invokes a method from Google's Web Services. Let's rock and roll.



Pic. 1: Exchanging SOAP packets between the client and Google Web Services

The skeleton of our client code (pseudo codes) will look something like following (Pic. 2):

```
<?php
```

```
if (!$_POST['query'])
```

```
{
```

```
//display form
```

Search		
Search term	Bluebonnet	

```
}
```

```
else
```

```
{
```

```
//execute query on Google
```



```
}
```

Pic. 2: Code Skeleton

As you can see, the script is split into two sections, one for the search form and the other for the search result. An "if" statement is used to decide which section of the script to execute.

Page 3: Walkthrough Example

```
<html>

<head><basefont face="Verdana" size="2"></head>

<body>

<?php

if (!$_POST['queryStr'])

{

?>

<h2>MyGoogle Search Engine</h2>

<form action="<?=$_SERVER['PHP_SELF']?>" method="post">

Type search term: <input type="text" name="queryStr">
```

```
</form>

<?
}

else

{

    // include the class from NuSOAP

    include("nusoap.php");

    // create a instance of the SOAP client object

    $soapclient = new soapclient("http://api.google.com/search/beta2");

    // uncomment the next line to see debug messages

    // $soapclient->debug_flag = 1;

    // prepare an array of input parameters to be passed to the remote procedure

    // doGoogleSearch()

    $params = array(

        'Googlekey' => 'gs8f1fJQFHJfBmgmratlW5z3nTQV0ts8', // Google license

        // key

        'queryStr' => $_POST['queryStr'], // search term that was being typed

        'startFrom' => 0, // start from result n

        'maxResults' => 10, // show a total of 10 results

        'filter' => true, // remove similar results

        'restrict' => '', // restrict by topic

        'adultContent' => true, // remove adult links from search result

        'language' => '', // restrict by language

        'iencoding' => '', // input encoding
```

```

        'oencoding' => ''                // output encoding

    );

/* invoke the method on the Googles server. The call() method accept four arguments- name of
the remote procedure to be invoked, an array of arguments for the remote procedure, namespace
and SOAP action */

$MyResult = $soapclient->call("doGoogleSearch", $params, "urn:GoogleSearch",
"urn:GoogleSearch");

/* Uncomment next line, if you want to see the SOAP envelop, which is forwarded to Google
server, It is important to understand the content of SOAP envelop*/

// echo '<xmp>'.$soapclient->request.'</xmp>';

/* Uncomment next line, if you want to see the SOAP envelop, which is received from Google
server. It is important to understand the SOAP packet forwarded from Google Server */

// echo '<xmp>'.$soapclient->response.'</xmp>';

// Print the results of the search

if ($MyResult['faultstring'])

{

?>

<h2>Error Report</h2>

<? echo $MyResult['faultstring'];?>

<?

}

else

{

```

```

?>

<h2>MyGoogle Search Results</h2>

Your search for <b><?=$MyResult['searchQuery']?></b> produced <?=$MyResult
['estimatedTotalResultsCount']?> hits.

<br>

<?  $i=1;

    if (is_array($MyResult['resultElements']))

    { echo "<table border=0 cellpadding=2 cellspacing=2>";

        foreach ($MyResult['resultElements'] as $r)

        {

            echo "<tr><td>[$i] <a href=" . $r['URL'] . ">" . $r['title'] . "</a>";

            echo $r['snippet'] . "(" . $r['cachedSize'] . ")</td></tr>";

            $i++;

        }

    }

    $i=1;

?>

</table>

<?

}

}

?>

</body>

</html>

```

Page 4: Conclusion

In this article, we have seen how developers write software programs that connect remotely to the Google's Web Services. Communication is performed via the Simple Object Access Protocol (SOAP),

an XML-based mechanism for exchanging typed information. Java and .NET programmers may find an example [here](#). It is also possible to write equivalent code using the [PEAR SOAP](#) client.

The example in this article has been tested on Windows XP server and Linux/PIII with Apache 1.3.24 and PHP 4.2.1 and PHP 4.3.

If you have any questions or comments about this article please post them [here](#).

For more great programming articles, please visit <http://www.devarticles.com>. If you have an opinion or question about this article, then please post it at the devArticles developer forum, which is located at <http://www.devarticles.com/forum>