

Integrating Google search into your site using ColdFusion and XML

Posted At : 12 March 2010 15:13 | Posted By : Shaun McCran

Related Categories: Google, Development, Coldfusion

Rather than writing a custom search, and indexing service for your sites you can use Google to do the leg work for you.

This article examines how you can create and integrate a Google site search into your site. We will query Google and return an XML packet, which we will translate and display within our sites framework.

I am using XML as I must parse the result myself for Accessibility. There are a few different ways of returning result, including some really nice AJAX stuff. None of them are Accessable for disabled users though.

To start with go here: <http://www.google.co.uk/sitesearch>

You can setup a free or Business account. This allows you to specify which sites the search will index, and any exclusions or filtering rules you want. The main difference between the free and Business accounts is the control you have over advertising.

The first step to create a simply search form, the simple form below will submit a form value to a processing template that we will build below.

```
<form name="gSearch" action="page.cfm" method="post">
<label for="q">Search:</label> <input type="text" name="q" id="q">
</form>
```

Next we need to create the template that will handle the form value, and process the results (page.cfm from the form above). The first step is to

create a url string that we will use in a cfhttp request. This fires your form value to Google. We also specify that we want the return in an XML format. The http request is parsed using xmlParse.

```

        <cfparam name="attributes.q" default="">
        <h1>Search results</h1>

        <cfif not len(attributes.q)>
        Please enter some search text
        </cfif>

.url = "http://www.google.com/cse?cx=YOUR_ACCOUNT_NUMBER&client=google-csbe&output=xml"
        <cfset variables.url = variables.url & attributes.q>

<cfhttp url="#arguments.url#" method="get" result="variables.response"></cfhttp>

        <cfscript>
        xmlfile = xmlparse(variables.response.filecontent);
        </cfscript>

        <cftry>
        <cfset arrEntries = variables.response.gsp.res.r>
        <cfcatch>
        <!-- no response -->
        <cfset variables.noResults = true>
        </cfcatch>
        </cftry>
    
```

In the next block of code we check that we successfully parsed the XML and that we have a valid Array. Then we loop over the results Array setting the values we want. Then display them on the page.

```

        <cfif variables.noResults>
<p>There were no results for your search using <strong>#attributes.q#</strong></p>
        </cfif>
        <cfelse>
<p>Below are the results for your search on '<strong>#attributes.q#</strong>'</p>
        <p></p>

        <cfloop index="i" from="1" to="#ArrayLen(arrEntries)#">
        <cfset properties["searchTitle"] = arrEntries[i].XmlChildren[3].XmlText>
        <cfset properties["searchString"] = arrEntries[i].XmlChildren[5].XmlText>
        <cfset properties["searchUrl"] = arrEntries[i].XmlChildren[1].XmlText>
        <a href="#properties.searchUrl#" title="#properties.searchTitle#" target="_blank">#pr
    
```

```
<div>#properties.searchString#</div>  
    <br>  
  </cfloop>  
  </cfif>  
</cfif>
```

The returned XML packet has a lot more data in it than just the three values I am using, so you could expand on this.