Geo coding Latitude and Longitude address in coldfusion using CFhttp

Posted At : 29 June 2009 17:32 | Posted By : Shaun McCran
Related Categories: Google, Development, Coldfusion, Web technologies

One piece of recently functionality to a site I'm writing is the ability to look up places on a Google powered map.

There are a variety of ways to insert a Google map into your site, but the first real hurdle is the lookup code.

Google does not use an address to position its map, it uses the Latitude and Longitude co-ordinates to place the map area around the desired location.

Google has pretty extensive documentation around this here:

http://code.google.com/apis/maps/documentation/geocoding/index.html

Rather than translate the locations on the fly on a per-hit basis I thought I would perform the lookup when the record is submitted to the database, that way I can cut down the number of google hits, and just reference the local data. Google also prefers this method, as it is less process intensive on their end of things.

First you need an API key:

http://code.google.com/apis/maps/signup.html

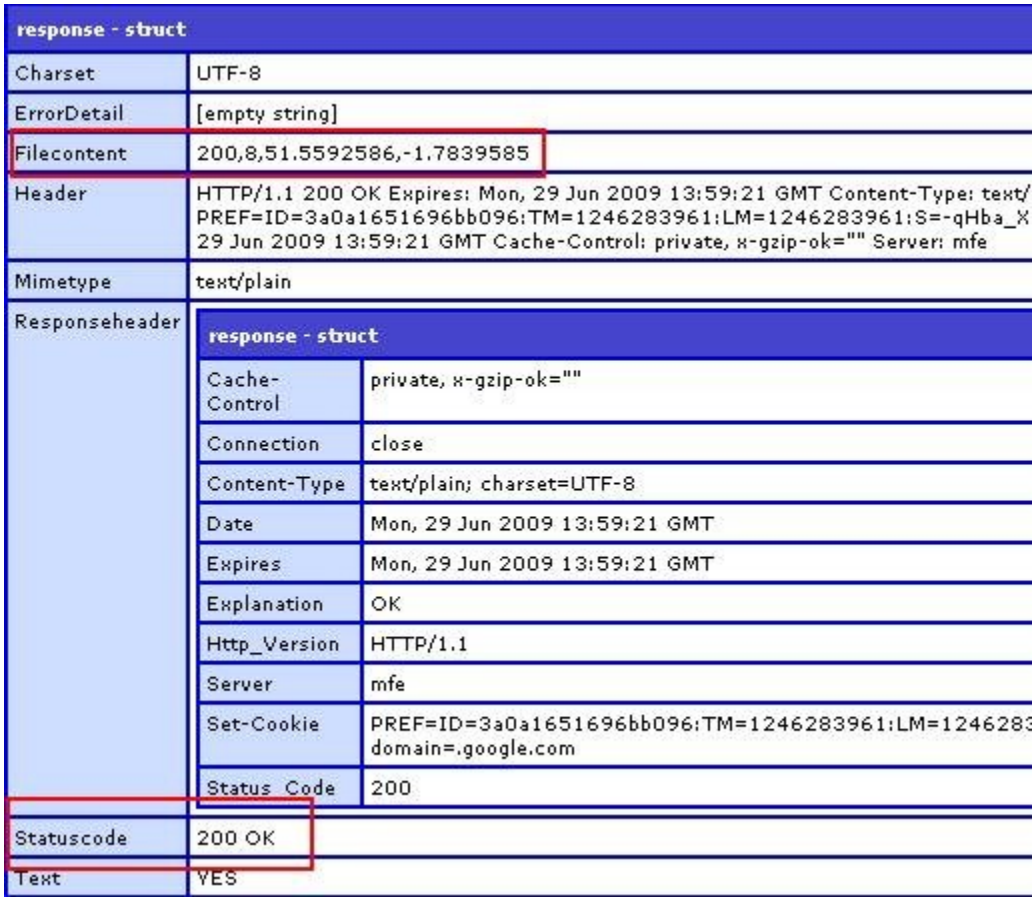
This application already has methods for setting the data in a table, so I am simply going to call another packaged method to calculate the latitude and longitude, and store them in the table with the other data.

```
lat long for an address: docs at http://code.google.com/apis/maps/documentation/ge
="address" displayName="Address to Geo" type="string" hint="String of the address
```

```
                        <cfset var geoDetails = structNew()>

                      <cfset var apiKey = "Your API key here">

                              <!--- initial string --->
            <cfset var requestString = "http://maps.google.com/maps/geo?">

                           <!--- q= address to geo code --->
    <cfset requestString = requestString & "q=28+Morley+Street,Swindon,SN1+1SG

                           <!--- key = API key --->
             <cfset requestString = requestString & "key=" & apiKey & "&">

           <!--- sensor = does the requestor have a location sensor? --->
            <cfset requestString = requestString & "sensor=false" & "&">

                        <!--- output = output format --->
             <cfset requestString = requestString & "output=csv" & "&">

                       <!--- oe = output encoding format --->
              <cfset requestString = requestString & "oe=utf8" & "&">

                       <!--- gl= Country code pointer --->
                 <cfset requestString = requestString & "gl=uk">

       <cfhttp url="#requestString#" method="get" result="response"></cfhtt

               <!--- returns 4 elements statuscode/accuracy/lat/long
                              Higher accuracy is better  --->
         <cfset geoDetails.status = listGetAt(response.filecontent,'1',',')>
        <cfset geoDetails.accuracy = listGetAt(response.filecontent,'2',',')
          <cfset geoDetails.lat = listGetAt(response.filecontent,'3',',')>
          <cfset geoDetails.long = listGetAt(response.filecontent,'4',',')>

                             <cfreturn geoDetails />
                              </cffunction>
```

As you can see from above, I am simply creating a text string URL, and using cfhttp to GET the result from http://maps.google.com/maps/geo?

The screenshot below show the returned responses, and the http status code.



response - struct	
Charset	UTF-8
ErrorDetail	[empty string]
Filecontent	200,8,51.5592586,-1.7839585
Header	HTTP/1.1 200 OK Expires: Mon, 29 Jun 2009 13:59:21 GMT Content-Type: text/ PREF=ID=3a0a1651696bb096:TM=1246283961:LM=1246283961:S=-qHba_X 29 Jun 2009 13:59:21 GMT Cache-Control: private, x-gzip-ok="" Server: mfe
Mimetype	text/plain
Responseheader	response - struct
Statuscode	200 OK
Text	YES

response - struct	
Cache-Control	private, x-gzip-ok=""
Connection	close
Content-Type	text/plain; charset=UTF-8
Date	Mon, 29 Jun 2009 13:59:21 GMT
Expires	Mon, 29 Jun 2009 13:59:21 GMT
Explanation	OK
Http_Version	HTTP/1.1
Server	mfe
Set-Cookie	PREF=ID=3a0a1651696bb096:TM=1246283961:LM=124628 domain=.google.com
Status_Code	200

The result is parsed into a struct and returned to the parent function to be stored. Far less overhead than doing this for every map call.

Please note that this is far more heavily commented for Blog purposes. Now to actually call the service using the lat and long variables stored, but thats another article.