

Arc en Ciel Ltd.

Gazetteer Webservice FactSheet

Overview

We provide two gazetteer webservices: on place name and on street name.

The place name service allows a user to browse for any town, village or locality in the UK, including Northern Ireland. Over 60,000 place names are present, covering even the smallest settlements. Also included are railway, London underground and DLR stations and airports.

The street name service contains almost 900,000 separate streets in the UK, including Northern Ireland. It is possible to restrict browsing to an area around a given centre, so that only local streets will be returned.

Place Name Gazetteer

To see how the place name service works, simply paste the following link into the address field of your browser (you'll have to do it in two parts). The clearest results are given in Internet Explorer, as Firefox and Google Chrome don't display XML tags:

www.bifrost.co/webservices/gazetteer.aspx?user=test&password=password&function=placelist&text=st.albans

You will see the result:-

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Results for: st. albans -->
<Results>
  <PLACE>
    <NAME>St. Albans,Hertfordshire</NAME>
    <TYPE>L</TYPE>
    <LNG>-0.3347</LNG>
    <LAT>51.7513</LAT>
    <GRE>515051</GRE>
    <GRN>207163</GRN>
  </PLACE>
</Results>
```

You now have the full name and county for the place. You also have an estimate of its size: 'L' is for 'Large Settlement' – over 10,000 residents. See the Data Structure section for a full list of possible types.

The map reference of the centre of the place is also given as both a longitude/latitude pair and an Ordnance Survey grid reference.

The above example is where the place name is unique. If you put in, say, Springhill, you'll get eleven results, including:-

Springhill, Abingdon, Oxfordshire
Springhill, East Cowes, Isle of Wight
Springhill, Glasgow, (Lanarkshire)
Springhill, Lichfield, Staffordshire
Springhill, Moreton-In-Marsh, Gloucestershire
(etc)

You will normally arrange for the webservice to populate a listbox, from which the user will see the Post Town and county and be able to select the correct alternative.

Street Name Gazetteer

The street name webservice works in a similar fashion to the place name service. To see it in operation, paste the following into your browser (you'll have to do it in two parts). The clearest results are given in Internet Explorer, as Firefox and Google Chrome don't display XML tags:

[www.bifrost.co/webservices/gazetteer.aspx?user=test&password=password&function=streetlist&text=station arcade](http://www.bifrost.co/webservices/gazetteer.aspx?user=test&password=password&function=streetlist&text=station%20arcade)

You will see the result:-

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Results for: station arcade -->
<Results>
  <STREET>
    <NAME>Station Arcade, Great Portland Street, London, (Greater London)</NAME>
    <LOCATION>
      <PCD>W1W 5PW</PCD>
      <LNG>-0.1439</LNG>
      <LAT>51.5237</LAT>
      <GRE>528861</GRE>
      <GRN>182174</GRN>
    </LOCATION>
  </STREET>
</Results>
```

The result gives all the streets which have a name beginning with "station arcade". In this case only one, which is very unusual. Many names such as "Station Road" will potentially return hundreds of results. For this reason, the webservice is able to filter results by position. Taking the longitude and latitude of the centre of St. Albans as returned in the place name example, and using a radius of 5km, we can create the URL:-

[www.bifrost.co/webservices/gazetteer.aspx?user=test&password=password&function=streetlist&text=station road¢re=-0.3347,51.7513&radius=5000](http://www.bifrost.co/webservices/gazetteer.aspx?user=test&password=password&function=streetlist&text=station%20road¢re=-0.3347,51.7513&radius=5000)

Now we get a single result – Station Road,Smallford,St Albans,Hertfordshire.

In many cases you will want to “hardwire” the centre, if for example the website is for a local taxi service. In other circumstances you might want the user to select a town first, then perform a search for a street.

Following the street name is a list of all postcodes in the street, with the map reference of the centre of each postcode. The map reference is given as both a longitude/latitude pair and an Ordnance Survey grid reference. In the above example there is only one postcode, but most streets have two or three postcodes and some have dozens.

Search String Considerations

The search string is not case-sensitive.

There is a special case in that if the string starts with “St.” this is expanded to “Saint” for sorting purposes.

Non-alphanumeric characters are considered equivalent to spaces, so “Ross-on-Wye” is equivalent to “Ross on Wye”.

Single quotes cannot be sent in a URL string. For this reason, they are ignored for sorting purposes and your code should remove them from user-typed input.

JavaScript Object Notation (JSON)

The results shown so far have been in XML format, which is widely used. However, for users of JavaScript, it is also possible to produce output in JSON notation. To get this add the parameter “[output=JSON](#)” to the calling URL. You may also add “[callback=<fname>](#)” to supply a callback function name. If you omit this the name “[LoadJSON](#)” will be used.

The equivalents of the above examples are then:-

```
LoadJSON (
{
  "Results":
  {
    "PLACE":
    [
      {
        "NAME": "St. Albans,Hertfordshire",
        "TYPE": "L",
        "LNG": "-0.3347",
        "LAT": "51.7513",
        "GRE": "515051",
        "GRN": "207163"
      }
    ]
  }
}
)
```

And:-

```
LoadJSON (
```

```

{
  "Results":
  {
    "STREET":
    [
      {
        "NAME": "Station Arcade,Great Portland Street,London,(Greater
London)",
        "LOCATIONS":
        [
          {
            "PCD": "W1W 5PW",
            "LNG": "-0.1439",
            "LAT": "51.5237",
            "GRE": "528861",
            "GRN": "182174"
          }
        ]
      }
    ]
  }
}
)

```

If this is sent to a function as LoadJSON(data) then each name can be loaded with lines like:-

```
var placename = data.Results.PLACE[i];
```

or:-

```
var streetname = data.Results.STREET[i];
```

Implementation

Javascript

This is the easiest way to add gazetteer lookup to a webpage. A complete HTML file is given below. It works by encapsulating the URL string into a script element and appending it to the document. This issues the URL. The reply is a function call which is received by LoadJSON. This parses the data into a selection list.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Place Name Gazetteer Demonstration</title>

<script type="text/javascript">
function getlocation()
{
    // Load place name (remove single quotes)
    var placename=document.getElementById('txtplacename').value;
    placename = placename.replace(/'/g, '');

    // Create a script element that will issue the URL
    var loader = document.getElementById('jsonp_loader');
    if (!(loader == null)) loader.parentNode.removeChild(loader);
    var script = document.createElement("script");
    script.setAttribute("id", "jsonp_loader");
    var url =
"http://www.bifrost.co/webservices/gazetteer.aspx?user=test&password=
password&function=placelist&output=JSON&text=" + placename;
    script.setAttribute("src", url);

    // Append the element to the document, which will issue the URL.
    // The reply will call the LoadJSON function
    document.getElementsByTagName("head")[0].appendChild(script);
}

function LoadJSON(data)
// This is the callback function that will receive the reply from
// the webservice. Its name is the default. You can change this
// using the callback=<function name> parameter
{
    // Clear any existing results
    document.getElementById('lstplaces').innerHTML = '';

    // Check for error message
    if (data.Result != null) alert(data.Result.ERRORMESSAGE);
    else
    {
        // Take each place name in turn
        for (var i=0; i<data.Results.PLACE.length; i++)
        {
            // Create an Option and add it to the Selection List

            var opt = document.createElement("option");
            var item = data.Results.PLACE[i];
```

```
        opt.setAttribute('value', item.LNG + "|" + item.LAT);
        opt.text = item.NAME;
        document.getElementById('lstplaces').appendChild(opt);
    }
}
</script>
</head>

<body>
<input id="txtplacename" type="text"/>
<input type="button" onclick="getlocation()" value ="Get Place"/>
<br />
<select id="lstplaces" multiple='multiple' size='10'
style="width:300px"></select>
</body>
</html>
```

ASP.NET with Visual Basic

Put the following declarations at the head of the source file.

```
Imports System.NET
Imports System.IO
```

Then to perform a lookup use:-

```
' Send the starting characters of a place name (remove single quotes)
Dim sPlace As String = "st.a"
sPlace = sPlace.Replace("'", "")
Dim sURL As String =
    "http://www.bifrost.co/webservices/gazetteer.aspx?user=test&" +
    "password=password&function=placelist&text=" + sPlace

' Read the response from the website into a string
Dim wc As WebClient = New WebClient()
Dim arr As Byte() = wc.DownloadData(sURL)
Dim sXML As String = ASCIIEncoding.ASCII.GetString(arr)

// sXML now holds the results in XML format
```

ASP.NET with C#

Put the following declarations at the head of the source file:-

```
using System.Net;
using System.Text;
```

To perform a lookup use:-

```
// Set the lookup string with a test postcode
string sPostcode = "cb11du";
string sURL =
    "http://www.bifrost.co/webservices/gazetteer.aspx?user=test&" +
    "password=password&function=placelist&text=" + sPlace

// Read the response from the website into a string
WebClient wc = new WebClient();
byte[] arr = wc.DownloadData(sURL);
string sXML = ASCIIEncoding.ASCII.GetString(arr);

// sXML now holds the results in XML format
```

PHP

In PHP the code is simply:-

```
// Send the starting characters of a place name(remove single quotes)
$place = "st.a";
$place = str_replace("'", "", $place);
$url =
    "http://www.bifrost.co/webservices/gazetteer.aspx?user=test&" +
    "password=password&function=placelist&text=" + $place;

// Perform the lookup
$xmlResult = file_get_contents( $url );

// $xmlResult now holds the list of place names in XML format.
```

Data structure

Place List

The following data is supplied on place list searches for each matching Place. All fields will contain data.

Code	Description	Comment
NAME	Place name and address	
TYPE	Place name type code	'S' – Small Settlement: up to 1000 'M' – Medium Settlement: 1000-10,000 'L' – Large Settlement: over 10,000 'A' – Airport 'R' – Railway station 'U' – London Underground station 'D' – DLR station
LNG	Longitude	WGS84
LAT	Latitude	WGS84
GRE	Easting Grid Reference	An x-coordinate in metres
GRN	Northing Grid Reference	An y-coordinate in metres

Street List

The following data is supplied on street list searches for each matching Street. The Name field will always contain data. On rare occasions other fields may be blank.

Code	Description	Comment
NAME	Street name and address	
LNG	Longitude	WGS84
LAT	Latitude	WGS84
GRE	Easting Grid Reference	An x-coordinate in metres
GRN	Northing Grid Reference	An y-coordinate in metres
PCD	Postcode	

Error Messages

If an error occurs, the XML response will look something like:-

```
<?xml version="1.0" encoding="utf-8" ?>
<Result>
  <ErrCode>4</ErrCode>
  <ErrorMessage>Invalid user name or password</ErrorMessage>
</Result>
```

The JSON equivalent is:-

```
LoadJSON(
{
  "Results":
  {
    "ERRCODE": "4",
    "ERRMESSAGE": "Invalid user name or password"
  }
}
)
```

A complete list of all errors that may be returned is:-

1	Missing user name
2	Missing password
3	Missing/invalid function
4	Invalid user name or password
5	Missing search string
6	Invalid centre
7	Invalid radius
8	Invalid postcode
9	Only search strings beginning with 'S' allowed in testing
100	System error: couldn't open data file
101	System error: couldn't open index file
102	System error: duplicate user
103	System error: invalid credit check

Pricing

When you first purchase credits, you will be assigned a user name and password. These will be used in the web access. You can supply an email address and a top-up point. When your credits reduce to this level, you'll automatically be sent a message inviting you to top up.

To find your credit balance at any time, use: “[function=getcredit](#)”.

Unused credits expire a year after the date of purchase.

Most users will find the Pay As You Go solution the most economical. However, if your website is very heavily used you can get unlimited use for a fixed cost.

Full pricing can be seen on our website at www.arcenciel.com/gazetteer or contact us directly to discuss your requirements.