Pieter De Rycke
Bio | Website | @PieterDeRycke

Using Google Geocode to get GPS coordinates from an address

02.22.2012 | 8377 views | J'aime 5 Tweet 2 0 Share 2

For a hobby project, I recently needed to calculate distances between stores and find nearby stores. In order to implement the required functionality, I have used the Google Geocoding service to have GPS coordinates for all my database entries and then it was just a matter of applying the correct math.

The Google Geocoding service is a REST service offered free of charge by Google and no developer sign-up is required. It can translate an address string into GPS coordinates. Result data can be returned in XML or JSON format. The only limitation is that the free version can only geocode 2500 addresses per day.

More technical information can be found on the following web page: http://code.google.com/intl/en-US/apis/maps/documentation/geocoding.

In this blog post, I want to share the tinny wrapper I have created to simplify calling this web service.

I first created a generic interface for the Geocoder.

```
1. public interface IGeocoder
2. {
3.    Coordinates Geocode(string address);
4. }
```

I also created a small data structure for my coordinates.

```
01. public class Coordinates
02. {
03.    public Coordinates(double latitude, double longitude)
04.    {
05.       Latitude = latitude;
06.       Longitude = longitude;
07.    }
08.
09.    public double Latitude { get; private set; }
10.
11.    public double Longitude { get; private set; }
12. }
```

Then I wrote the following implementation that uses the Google web service. I use the XML format for geocoded addresses because XML is easier to parse in .Net applications without requiring 3rd party libraries. If you want to use this service from a WP7 application it would be better to use the

more compact JSON format.

```
01. public class GoogleGeocoder : IGeocoder
02. {
03.    private const string ServiceUri = "http://maps.googleapis.com
          /maps/api/geocode/xml?address={0}&region=be&sensor=false";
04.
05.    public Coordinates Geocode(string address)
06.    {
07.       if (string.IsNullOrEmpty(address))
08.          throw new ArgumentNullException("address");
09.
10.       string requestUriString = string.Format(ServiceUri,
             Uri.EscapeDataString(address));
11.
12.       HttpWebRequest request =
             (HttpWebRequest)HttpWebRequest.Create(requestUriString);
13.
14.       try
15.       {
16.          WebResponse response = request.GetResponse();
17.
18.          XDocument xdoc = XDocument.Load(response.GetResponseStream());
19.
20.          // Verify the GeocodeResponse status
21.          string status =
                xdoc.Element("GeocodeResponse").Element("status").Value;
22.          ValidateGeocodeResponseStatus(status, address);
23.
24.          XElement locationElement =
                xdoc.Element("GeocodeResponse").Element("result").Element("geo
25.          double latitude = (double)locationElement.Element("lat");
26.          double longitude = (double)locationElement.Element("lng");
27.
28.          return new Coordinates(latitude, longitude);
29.       }
30.       catch (WebException ex)
31.       {
32.          switch(ex.Status)
33.          {
34.             case WebExceptionStatus.NameResolutionFailure:
35.                throw new ServiceOfflineException("The Google Maps
                      geocoding service appears to be offline.", ex);
36.             default:
37.                throw;
38.          }
39.       }
40.
41.    }
42.
43.    private void ValidateGeocodeResponseStatus(string status, string
          address)
44.    {
45.       switch (status)
46.       {
47.          case "ZERO_RESULTS":
48.             string message = string.Format("No coordinates found for
                   address \"{0}\".", address);
49.             throw new UnknownAddressException(message);
50.          case "OVER_QUERY_LIMIT":
51.             throw new OverQueryLimitException();
52.          case "OK":
53.             break;
54.          default:
55.             throw new Exception("Unkown status code: " + status + ".");
56.       }
57.    }
58. }
```

References

Reference:

Translating an address to GPS coordinates with the Google Geocoding REST service

Published at DZone with permission of Pieter De Rycke, author and DZone MVB. (source)

(Note: Opinions expressed in this article and its replies are the opinions of their respective authors and not those of DZone, Inc.)

Tags: geogode Google GPS how-to service Tools

Asp.net: Cache entire gridview

What is the Need of Prism and Composite Applications in .NET?

How to debug silent crashes in .Net

See more popular at DZone

Subscribe to the RSS feed

Comments

Ajya Chang replied on Wed, 2012/02/29 - 1:48am

Hello,

This is a wonderful article. Here I can see that you have explained it with code also. Can I straight away use it? Or I need to go t the reference link provided below for understanding it. THis can be use ful in many we applications the google geocode api. Thanks once again for the code. I will recomment this. Thanks

Login or register to post comments

Comment viewing options

Flat list - expanded | Date - oldest first | 30 comments per page | Save settings

Select your preferred way to display the comments and click "Save settings" to activate your changes.