

US Postal Service Address Information API and Google Maps API “Point-In-Polygon”

Mike Christensen

32-611-02

Summer 2011

Introduction

For my class project, I decided that I wanted to focus more on the web application aspect of the class than on creating maps. In fact, I kept the web application very basic and focused more on functionality than aesthetics. So I set out to combine two APIs into one web application, not knowing whether I could actually put it together.

Proposal

Input a street address. Have the Post Office Address Information API determine whether the address is valid (i.e. is an address where mail can be delivered) and format the address in the Post Office standardized format. Have the Google Maps API determine whether the address lies within a pre-determined boundary. Display whether the address is deliverable by the Post Office, display the address in Post Office standardized format, display whether the address lies within a pre-determined boundary, and have the Google Maps API map the address.

Background

In the Church of Jesus Christ of Latter-day Saints, the basic congregational unit is called a “ward.” Each ward has a geographic boundary, and members of the Church attend the ward of which boundary they live within. Additionally, in areas where there are large populations of single members, YSA (Young Single Adult) wards are established, which overlay multiple “non-single” wards and give singles an increased opportunity to mingle and get married. (As I am still single, it doesn’t always work! LOL) The boundary that I chose for my project is the boundary of the Rose Park YSA Ward, which is the ward that I live within and attend.

Step 1: The US Postal Service Address Information API

- Best Example: <http://zip4.usps.com/>
- Evaluates whether the address is deliverable.
- Returns the address formatted according to standardized format:
<http://pe.usps.com/text/pub28/welcome.htm>
- Requires signing up for API key:
<http://www.usps.com/webtools/address.htm>

Step 1: Hurdles!

- Requires passing XML to Postal Service API.
- And receiving XML from Postal Service API.
- Problem: API manual gives no examples.
- Solution: Search internet for an example.
- Found it here: <http://joe-riggs.com/blog/2009/10/address-standardization-verification-with-usps-web-tools-and-php/>

Step 2: Point-in-Polygon

- “Point-in-polygon” is the general GIS term for determining whether a point lies within a polygon.

Step 2: Hurdles!

- Problem: Google Maps API lacks a point-in-polygon function (along with other basic GIS functions).
- Solution: Search internet for an example.
- Found it here:
<https://github.com/tparkin/Google-Maps-Point-in-Polygon>

Step 2: More Hurdles!

- Problem: Point-in-polygon code requires formatting boundary as a JavaScript array.
- Solution: Search internet for an example.
- Found it here:
http://www.geocodezip.com/v3_polygon_example.html

Step 3: The Boundary

- Boundary of the Rose Park YSA Ward previously drawn using ArcMap.

Step 3: Hurdles!

- Problem: Boundary needs to be converted from Geodatabase feature class to JavaScript array.
- Solution:
 - Geodatabase feature class to KMZ via ArcToolbox
 - KMZ to KML via Google Earth
 - KML to single-line text string via Notepad++
 - Single-line text string to CSV via WordPerfect
 - CSV to formatted array via Excel
 - Formatted array to JavaScript array via Notepad++

Step 4: Putting It All Together

- The final product is a combination of HTML, PHP, and JavaScript.
- (For those not familiar with PHP, it is a “server-side” scripting language: <http://php.net>)

Step 4: Hurdles!

- Problem: Requires passing variables among HTML, PHP, and JavaScript.
- Solution: Search internet for an example.
- Found it here:
<http://www.skytopia.com/project/articles/comp sci/form.html>
- And general help with HTML, XML, PHP, and JavaScript here: <http://www.w3schools.com/>

Step 5: Handling Errors from the APIs

- Tested my web application using a variety of different address possibilities and ensured that the proper result was displayed.
- Test examples are listed at the bottom of page 1 of my web application.

Here's the URL:

<http://32611.cascadepeak.com/>

Feel free to let me know, if you have any
questions, comments, etc.

s254704@nwmissouri.edu

or

mrc@cascadepeak.com