Geocoding Services
Web Application Programming Interface (API)

Last updated 07/2015

_____

Contents

Geocoding Definition

Geocoding is the process of taking an address and returning an actual or calculated latitude/longitude coordinate. Depending on the parts of the address that are provided, determines to what granularity it is possible to geocode.

The current Geocoding Services engine requires a structure address be provided. The resulting lat/long is calculated along an address range.

There are two entry points for the geocoding service – single record submission and batch.

The acceptable input address parts are:

- Structure number and street name (required)
- City name (optional)
- State (optional)
- ZIP code (optional)

The single record service allows for all of these parts to be submitted in a single line, or as separate fields. The batch requires each field to exist (either with text or blank) in a delimited form, preceded by a unique ID.

Audience

This document is intended for application, website, and mobile developers within the U.S. Census Bureau and the general public who want to leverage the Geocoding Services capability.

This service is designed for coding a provided address, or file of addresses, to a latitude/longitude coordinate based on data that's been loaded into the geocoding engine from a MAF/TIGER benchmark database.

The optional inclusion of the Geographic Lookup (geoLookup) adds information to the result relating to various levels of geography that cover the afore mentioned latitude/longitude coordinate. GeoLookup results can also be obtained directly by searching on the latitude/longitude coordinates.

Single Record Geocoding Service Requests

A Geocoding Service API request must be in the following form:

http://geocoding.geo.census.gov/geocoder/*returntype /searchtype*?parameters

Required Parameters

- `returntype` – **locations** (to get just geocoding response) or **geographies** (to get geocoding response as well as geoLookup)
- `searchtype` – **onelineaddress** OR **address** OR **coordinates**
- `benchmark` – A numerical ID or name that references what version of the locator should be searched. This generally corresponds to MTDB data which is benchmarked twice yearly. A full list of options can be accessed at http://geocoding.geo.census.gov/geocoder/benchmarks. The general format of the name is *DatasetType_SpatialBenchmark*. The valid values for these include:
    - *DatasetType*
        - *Public_AR*
    - *SpatialBenchmark*
        - *Current*
        - *ACS2015*
        - *Census2010*

  So a resulting benchmark name could be "Public_AR_Current", "Public_AR_Census2010", etc. Over time, there will always be a "Current" benchmark. It will change as the underlying dataset changes.
- `vintage` – a numerical ID or name that references what vintage of geography is desired for the geoLookup (only needed when returntype = geographies).). A full list of options for a given benchmark can be accessed at http://geocoding.geo.census.gov/geocoder/vintages?benchmark=*benchmarkId.* The general format of the name is *GeographyVintage_SpatialBenchmark.* The SpatialBenchmark variable should always match the same named variable in what was chosen for the benchmark parameter. The GeographyVintage can be Current, ACS2012, etc. So a resulting vintage name could be "ACS2015_Current", "Current_Census2010", etc. Over time, there will always be a "Current" vintage. It will change as the underlying dataset changes.
- `address (searchtype = onelineaddress)` – A single line containing the full address to be searched
  OR
- `street, city, state, zip (searchtype = address)` – The address split into the parts indicated. Not all parts need to be specified.
- `x, y (searchtype = coordinates)` – The latitude and longitude represented as decimal x/y values. Only returns geoLookup data. Can only be used with returntype = geographies.

Optional Parameters

- `format` – The format to be used for returning the standardized output (json, html).
- `layers` – By default, State, County, Tract, and Block layers are displayed when "geographies" is the chosen returntype. If additional or different layers are desired, they can be specified in a comma delimited list by ID or name as listed in the TigerWeb WMS layers, for instance here:

https://tigerweb.geo.census.gov/ArcGIS/rest/services/TIGERweb/tigerWMS_Current/MapServer

a valid entry could be: layers=14,16,18 OR layers=Unified School Districts,Secondary School Districts,Elementary School Districts. Only layers without the word "Labels" are considered. If all layers are desired, layers=all is an accepted entry.

In cases where the SpatialBenchmark selected is Census2010, the TIGERweb WMS needed is:

https://tigerweb.geo.census.gov/ArcGIS/rest/services/Census2010/tigerWMS_Census2010/MapServer

Geocoding Service Responses

Geocoding Services responses are returned in the format indicated by the `format` parameter value in the URL request's path.

JSON Output Format

In this example, Geocoding Services API requests a `json` response for the address "4600 Silver Hill Rd, Suitland, MD 20746":

http://geocoding.geo.census.gov/geocoder/locations/onelineaddress?address=4600+Silver+Hill+Rd%2C+Suitland%2C+MD+20746&benchmark=9&format=json

Other valid examples:

http://geocoding.geo.census.gov/geocoder/locations/address?street=4600+Silver+Hill+Rd&city=Suitland&state=MD&zip=20746&benchmark=Public_AR_Census2010&format=json

http://geocoding.geo.census.gov/geocoder/locations/address?street=4600+Silver+Hill+Rd&city=Suitland&state=MD&benchmark=9&format=json

The JSON returned by this request is shown below. Note the actual JSON may contain less whitespace. You should not make assumptions about the amount or format of whitespace in requests.

```
{
  "result": {
    "input": {
      "address": {
        "address": "4600 Silver Hill Rd, Suitland, MD 20746"
      },
      "benchmark": {
```

```
      "id": 9,
      "benchmarkName": "Public_AR_Census2010",
      "benchmarkDescription": "Public Address Ranges – Census 2010",
      "isDefault": false
    }
  },
  "addressMatches": [
   {
      "matchedAddress": "4600 Silver Hill Rd, SUITLAND, MD, 20746",
      "coordinates": {
       "x": -76.926956,
       "y": 38.846565
      },
      "tigerLine": {
       "tigerLineId": "613199520",
       "side": "L"
      },
      "addressComponents": {
       "fromAddress": "4600",
       "toAddress": "4712",
       "preQualifier": "",
       "preDirection": "",
       "preType": "",
       "streetName": "Silver Hill",
       "suffixType": "Rd",
       "suffixDirection": "",
       "suffixQualifier": "",
       "city": "SUITLAND",
       "state": "MD",
       "zip": "20746"
      }
    }
  ]
 }
}
```

Example using geoLookup:

http://geocoding.geo.census.gov/geocoder/geographies/address?street=4600+Silver+Hill+Rd&city=Suitland&state=MD&benchmark=Public_AR_Census2010&vintage=Census2010_Census2010&layers=14&format=json

Additional JSON response data for geoLookup:

```
"geographies":
        {"Census Blocks":
        [{"BLKGRP":"1",
        "STATE":"24",
```

```
"FUNCSTAT":"S",
"AREAWATER":0,
"NAME":"Block 1084",
"SUFFIX":"",
"LSADC":"BK",
"CENTLON":"-076.9266698",
"HU100":0,
"BLOCK":"1084",
"BASENAME":"1084",
"INTPTLAT":"+38.8477828",
"POP100":0,
"MTFCC":"G5040",
"COUNTY":"033",
"GEOID":"240338024051084",
"CENTLAT":"+38.8477828",
"INTPTLON":"-076.9266698",
"AREALAND":33460,
"OBJECTID":4385889,
"TRACT":"802405"
}]
}
```

If a JSONP-formatted result is desired/required, the format=`jsonp` can be specified with a corresponding Javascript callback method on the request.

Batch Geocoding

Geocoding can be accomplished in batch mode with submission of a .CSV, .XLS, or .XLSX formatted file. The file needs to be included as part of the HTTP request.

The file must be formatted in the following way:

Unique ID, Street address, City, State, ZIP

If a component is missing from the dataset, it must still retain the delimited format with a null value. Unique ID and Street address are required fields.

If there are commas that are part of one of the fields, the whole field needs to be enclosed in quote marks for proper parsing.

There is currently an upper limit of 1000 records per batch file.

The URL is as follows:

http://geocoding.geo.census.gov/geocoder/*returntype*/addressbatch

Required Parameters

- `returntype` – **locations** (to get just geocoding response) or **geographies** (to get geocoding response as well as geoLookup). Independent geoLookup ("coordinates" above) is not currently an available batch option.

- `benchmark` – A numerical ID or name that references what version of the locator should be searched. This generally corresponds to MTDB data which is benchmarked twice yearly. A full list of options can be accessed at http://geocoding.geo.census.gov/geocoder /benchmarks. The general format of the name is *DatasetType_SpatialBenchmark*. The valid values for these include:
  - *DatasetType*
    - *Public_AR*
  - *SpatialBenchmark*
    - *Current*
    - *ACS2015*
    - *Census2010*

  So a resulting benchmark name could be "Public_AR_Current", "Public_AR_Census2010", etc. Over time, there will always be a "Current" benchmark. It will change as the underlying dataset changes.

- `vintage` – a numerical ID or name that references what vintage of geography is desired for the geoLookup (only needed when returntype = geographies).). A full list of options for a given benchmark can be accessed at http://geocoding.geo.census.gov/geocoder/vintages?benchmark=*benchmarkId.* The general format of the name is *GeographyVintage_SpatialBenchmark.* The SpatialBenchmark variable should always match the same named variable in what was chosen for the benchmark parameter. The GeographyVintage can be Current, ACS2012, etc. So a resulting vintage name could be "ACS2015_Current", "Current_Census2010", etc. Over time, there will always be a "Current" vintage. It will change as the underlying dataset changes.

- `addressFile` – An input of type "file" containing the addresses to be coded

Example Shell Script Submission

curl --form addressFile=@localfile.csv --form benchmark=9
http://geocoding.geo.census.gov/geocoder/locations/addressbatch --output geocoderesult.csv