

Geocoding Services

Web Application Programming Interface (API)

Last updated 04/2024

Contents

- Geocoding Definition 1
- Audience 1
- Single Record Geocoding Service Requests 1
 - Required Parameters 2
 - Optional Parameters 3
- Geocoding Service Responses 4
 - JSON Output Format 4
- Batch Geocoding 6
 - Required Parameters 6
 - Example Shell Script Submissions 7

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)
- Urbanization (optional)
- City name (optional)
- Municipio (optional)
- State (optional)
- ZIP code (optional)

Further guidance on what address parts are supported/required for each search type is provided later in the documentation.

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 aforementioned 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:

<https://geocoding.geo.census.gov/geocoder/returntype/searchtype?parameters>

Note: The above URL will not work on its own. The **returntype** and **searchtype** in the URL needs to be replaced with one of the values in the next section.

Required Parameters

- **returntype** – **locations** (to get just geocoding response) or **geographies** (to get geocoding response as well as geoLookup)
- **searchtype** – **onelineaddress** OR **address** OR **addressPR** OR **coordinates** (supported only by the **returntype** = **geographies**)
- **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:

<https://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*
 - *ACS####* (#### will be replaced by the most recent ACS survey year)
 - *Census####* (#### will be replaced by the most recent Census year)

A resulting benchmark name could be “Public_AR_Current”. 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

<https://geocoding.geo.census.gov/geocoder/vintages?benchmark=benchmarkId>

Note: *benchmarkId* in the example above needs to be replaced with a valid benchmark ID. Here is an example using the "Current" benchmark ID (4):

<https://geocoding.geo.census.gov/geocoder/vintages?benchmark=4>

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*, *ACS####*, etc. A resulting vintage name could be “Current_Current”. 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. The *onelineaddress* **searchtype** supports Puerto Rico addresses without Urbanization and Municipio, but it does not support Puerto Rico addresses with Urbanization and Municipio. Puerto Rico addresses with an Urbanization and Municipio submitted to this **searchtype** may result in a no match or match to a Puerto Rico address without an Urbanization.
- **street, city, state, zip** (**searchtype** = **address**) – The address split into the parts indicated. Not all parts need to be specified. At a minimum, the geocoder expects the

`street` and `zip` parameters OR the `street`, `city`, and `state` parameters. The address `searchtype` supports Puerto Rico addresses without Urbanization and Municipio, but it does not support Puerto Rico addresses with Urbanization and Municipio. Puerto Rico addresses with an Urbanization and Municipio submitted to this `searchtype` may result in a no match or match to a Puerto Rico address without an Urbanization.

- `street, urb, city, municipio, state, zip (searchtype = addressPR)` – The Puerto Rico address split into the parts indicated. Not all parts need to be specified. `zip` should begin with 006, 007, or 009. To submit a Puerto Rico address without an Urbanization the geocoder expects at a minimum the `street` and `zip` parameters OR the `street`, `city`, and `state` parameters. To submit a Puerto Rico address with an Urbanization the geocoder expects at a minimum the `street`, `urb`, and `municipio` parameters.
- `x,y (searchtype = coordinates)` – The longitude and latitude 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 (supported values include `json` and `jsonp`).
- `callback` – The name of the function to pass the JSON response to as a parameter. The `format` parameter should be set to `jsonp` when using `callback` to request a JSONP response. At this time the geocoder does not support CORS requests so the `callback` and `format` parameters should be used in the client to make calls to the geocoder.
- `layers` – By default, State, County, Tract, Block, Congressional Districts, State Legislative Districts, Census Designated Places, County Subdivisions, Incorporated Places, Combined Statistical Areas, Urban Areas, Urbanized Areas, and Urban Clusters layers are displayed when “geographies” is the chosen `returntype`. It is possible that not all layers listed will return as this depends on the selected vintage and if the geography has nationwide coverage. 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 for the "Current_Current" vintage:
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 `Census####`, the TIGERweb WMS needed can be found at the following URL in the `Census####` folder:

<https://tigerweb.geo.census.gov/arcgis/rest/>

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, Washington, DC 20233":

<https://geocoding.geo.census.gov/geocoder/locations/onlineaddress?address=4600+Silver+Hill+Rd%2C+Washington%2C+DC+20233&benchmark=4&format=json>

Other valid examples:

https://geocoding.geo.census.gov/geocoder/locations/address?street=4600+Silver+Hill+Rd&city=Washington&state=DC&zip=20233&benchmark=Public_AR_Current&format=json

<https://geocoding.geo.census.gov/geocoder/locations/address?street=4600+Silver+Hill+Rd&city=Washington&state=DC&benchmark=4&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, Washington, DC 20233"},
    "benchmark": {
      "isDefault": true,
      "benchmarkDescription": "Public Address Ranges - Current Benchmark",
      "id": "4",
      "benchmarkName": "Public_AR_Current"
    }
  },
  "addressMatches": [{
    "tigerLine": {
      "side": "L",
      "tigerLineId": "76355984"
    },
    "coordinates": {
      "x": -76.92748724230096,
      "y": 38.84601622386617
    },
    "addressComponents": {
      "zip": "20233",
      "streetName": "SILVER HILL",
      "preType": ""
    }
  ]
}
```

```

    "city": "WASHINGTON",
    "preDirection": "",
    "suffixDirection": "",
    "fromAddress": "4600",
    "state": "DC",
    "suffixType": "RD",
    "toAddress": "4700",
    "suffixQualifier": "",
    "preQualifier": ""
  },
  "matchedAddress": "4600 SILVER HILL RD, WASHINGTON, DC, 20233"
}
}
}

```

Examples using geoLookup:

https://geocoding.geo.census.gov/geocoder/geographies/address?street=4600+Silver+Hill+Rd&city=Washington&state=DC&benchmark=Public_AR_Current&vintage=Current_Current&layers=10&format=json

Additional JSON response data for geoLookup:

```

"geographies": {"Census Block Groups": [{
  "GEOID": "240338024052",
  "CENTLAT": "+38.8538877",
  "AREAWATER": 8728,
  "STATE": "24",
  "BASENAME": "2",
  "OID": "20890286109972",
  "LSADC": "BG",
  "FUNCSTAT": "S",
  "INTPTLAT": "+38.8507922",
  "NAME": "Block Group 2",
  "OBJECTID": 210878,
  "TRACT": "802405",
  "CENTLON": "-076.9390445",
  "BLKGRP": "2",
  "AREALAND": 3168458,
  "INTPTLON": "-076.9410126",
  "MTFCC": "G5030",
  "COUNTY": "033"
}]}

```

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

<https://geocoding.geo.census.gov/geocoder/locations/onlineaddress?address=4600+Silver+Hill+Rd%2C+Washington%2C+DC+20233&benchmark=4&format=jsonp&callback=test>

Batch Geocoding

Geocoding can be accomplished in batch mode with submission of a CSV or Excel formatted file. CSV formatted files should have an extension ending in .CSV, .TXT, .DAT, and Excel files should have an extension ending in .XLS, or .XLSX. The file needs to be included as part of the HTTP request.

The CSV formatted file must be formatted in the following way for all addresses including Puerto Rico addresses without an Urbanization:

Unique ID, Street address, City, State, ZIP

The batch mode also supports Puerto Rico addresses with an Urbanization. Puerto Rico addresses with Urbanization must be formatted in the following way:

Unique ID, Street address, Municipio, State, ZIP, Urbanization

The two address formats can be submitted together in the same file. Excel submissions should place each component in its own column.

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 10,000 records per batch file.

The URL is as follows:

<https://geocoding.geo.census.gov/geocoder/returntype/addressbatch>

Note: The above URL will not work on its own. The **returntype** in the URL needs to be replaced with one of the values in the next section.

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. Batch geoLookup only includes state, county, tract, and block code. Currently there is no way to request additional geographies in the geographies **returntype**.
- **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:

<https://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*
 - *ACS####* (#### will be replaced by the most recent ACS survey year)
 - *Census####* (#### will be replaced by the most recent Census year)

A resulting benchmark name could be “Public_AR_Current”. 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:

<https://geocoding.geo.census.gov/geocoder/vintages?benchmark=benchmarkId>

Note: *benchmarkId* in the example above needs to be replaced with a valid benchmark ID. Here is an example using the "Current" benchmark ID (4):

<https://geocoding.geo.census.gov/geocoder/vintages?benchmark=4>

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*, *ACS####*, etc. A resulting vintage name could be “Current_Current”. 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 Submissions

Locations Example:

```
curl --form addressFile=@localfile.csv --form benchmark=4
https://geocoding.geo.census.gov/geocoder/locations/addressbatch --output geocoderesult.csv
```

Geographies Example:

```
curl --form addressFile=@localfile.csv --form benchmark=4 --form vintage=4
https://geocoding.geo.census.gov/geocoder/geographies/addressbatch --output
geocoderesult.csv
```