

Introduction to Geocoding

I. Geocoding 101:

The NYS Geocoder is available for public use and offers multiple options for geocoding. This guide is intended to assist users with connecting to the geocoding service in multiple ways given the user's:

- Intent to geocode a single address vs. multiple or a batch of addresses;
- Availability of GIS software; and/or
- Proficiency with scripting languages and/or application development.

Geocoding is the use of technology and reference data to return a geographic coordinate when an address location is entered. Geocoding is used when lists of addresses need to be placed on a map, when an address is entered in an application to center a map or return information for that location, or any other time you have an address and a geographic coordinate is needed.

Addresses can be entered either manually or by bulk input from a database or other source. The geocoder then compares the entered address to a set of reference data and returns a coordinate pair and standardized address for each input address it is able to match. The GIS Program Office geocoder uses a series of combinations of reference data and configuration parameters to optimize both the likelihood of a match and the quality of the results.

With geocoding, there are several terms to be familiar with:

- **Geocoding** – the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map.
- **Batch Geocoding** – the process of converting multiple addresses at a time into geographic coordinates.
- **Reverse Geocoding** – the process of converting geographic coordinates into a human-readable address.

II. How to Geocode:

A. The Locators

The first step in geocoding is understanding and choosing the appropriate locator for the type of address information you have. Address information is best understood in the context of the work done by the GPO's Street and Address Maintenance (SAM) Program. The SAM Program maintains statewide authoritative street and address point databases to support Next Generation 9-1-1 and other stakeholder and customer activities through extensive state and local government partnerships, and the use of data maintenance tools. Data products are compliant with National Emergency Number Association (NENA) standards, are publicly available, and are the source of the NYS enterprise geocoding service.

From 2012 through 2015 the NYS GIS Program Office built out a detailed statewide address point database by improving the positional accuracy of the existing NYS Address Points, adding sub-addresses (e.g., apartment, building, suite, floor), and moving them into the NENA address standard.

Building and maintaining street and address point data at the high level needed for 9-1-1 dispatch ensures it is suitable for multiple applications. For example, the SAM databases are also being used to host a statewide geocoding service, as state and local agencies need to geocode addresses as point locations to support a variety of functions. This allows multiple state agencies, local governments and other entities to utilize this shared service to get consistent, accurate geocoding results. As part of this initiative the SAM Program Team is responsible for maintaining a set of composite locators to assist geocoding efforts.

There are three composite locators to choose from in the ShareGIS service environment, each of which is made up of a series of locators. The following descriptions may help in determining which geocoder to use. Users should be aware that the GIS Program Office has found a few limitations of the address locators and has made ESRI aware of these limitations. A full list of these limitations and current work arounds are available on the [NYS GIS Clearinghouse website](#).

1. Street and Address Composite Locator

The first composite locator, the “**Street and Address Composite**,” is preferably used when the user has all the proper components of an address, namely:

- Street Number;
- Street;
- City;
- State; and
- Zip code

The Street and Address Composite locator is made up of a set of locators which examine each of these address components and compiles their matches into the best fit solution for a resulting coordinate pair. It is through this method, the examination of each part of the address, that users can expect the highest likelihood of a high-quality return.

The individual locators which make up the Street and Address Composite locator are listed in the order in which they will be accessed along with a brief description of the locator's source data. These six locators will generate the majority of the results when geocoding addresses.

Locator Name	Source Data	Description
1A_SAM_AP_ZipName	SAM Address Points	SAM address points using the postal zip code name for the city name in the locator.
1B_SAM_AP_CTName	SAM Address Points	SAM address points. The city or town name is used for the city name in the locator.
1C_SAM_AP_PlaceName	SAM Address Points	SAM address points. The city name is populated using the NYS Villages and Indian Reservations, the Census Designated Places and Alternate Acceptable Zip Code Names from the USPS. These names do not exist everywhere so there will be a limited number of points in this locator.
3A_SS_ZipName	NYS Street Segments	NYS Street Segments dataset using the postal zip code name for the city name in the locator. The location is interpolated from an address range on the street segment. The city name can be different for the left and right sides of the streets.
3B_SS_CTName	NYS Street Segments	NYS Street Segments using the city or town name for the city name in the locator. The location is interpolated from an address range on the street segment.
3C_SS_PlaceName	NYS Street Segments	NYS Street Segments using an alternate place name for the city field. This field is populated using the NYS Villages and Indian Reservations, the Census Designated Places, and Alternate Acceptable Zip Code Names from the USPS. These areas do not exist everywhere so there will be a limited number of segments with this attribute. The location is interpolated from an address range on the street segment.

2. Street NoNum and ZipCode Composite Locator

Any address that does not successfully geocode to the Street and Address Composite can then be run through the second composite locator, the Street NoNum and ZipCode Composite. The intent of this composite locator is to still return a coordinate pair in the event a user does not know the street number or zip code for the address they are trying to match. **It is important for users to recognize that hits from this locator will not be spatially accurate.** This composite locator is made up of the following locators:

Locator Name	Source Data	Description
4A_SS_NoNum_ZipName	NYS Street Segments	NYS Street Segments dataset using the postal zip code name for the City name in the locator. The location is placed on a street segment with the matching name. Please note this may or may not be the correct street segment.
4B_SS_NoNum_CTName	NYS Street Segments	NYS Street Segments dataset using the city or town name is used for the city name in the locator. The location is placed on a street segment with the matching name. Please note this may or may not be the correct street segment.
4C_SS_NoNum_PlaceName	NYS Street Segments	NYS Street Segments dataset using the alternate place name is used for the city name in the locator. This field is populated using the NYS Villages and Indian Reservations, the Census Designated Places and Alternate Acceptable Zip Code Names from the USPS. These areas do not exist everywhere so there will be a limited number of segments with this attribute. The location is placed on a street segment with the matching name. Please note this may or may not be the correct street segment.
5_ZipCodePts	Zip Code boundaries	Point placed at the centroid of the Zip Code boundaries.

Lastly, there is also a locator available that will allow the user to find Incorporated and Unincorporated Places throughout New York State. This locator can be used when only a place name is known for an address.

3. NYPlace

Locator Name	Source Data	Description
NYPlace	Municipality Centroid Points	This locator contains points placed at the centroid of NYS Cities, Towns, Villages, Indian Reservations, Unincorporated Places, and Neighborhoods.

B. Locator Functionality

The locators can be accessed in multiple ways:

- Using GIS software and connecting to the geocoder via url;
- Through the use of custom generated scripts that make a request to a locator's rest endpoint;
- Directly via the ShareGIS services website ([https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street and Address Composite/GeocodeServer/findAddressCandidates](https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/findAddressCandidates)) where an easy-to-use interface is provided for address input and coordinate generation.

Specific functionality available for each locator may vary depending on whether an application is being used or whether the user is directly interacting with the geocoding services available from the ShareGIS services environment.

The discussion below investigates the more popular methods of interacting with the NYS geocoder and its composite locators, and as additional methods are discovered this document will be updated.

Integration Table of Contents

Hold down the “Control” button and Click on a method in the Table of Contents to jump to its integration details.

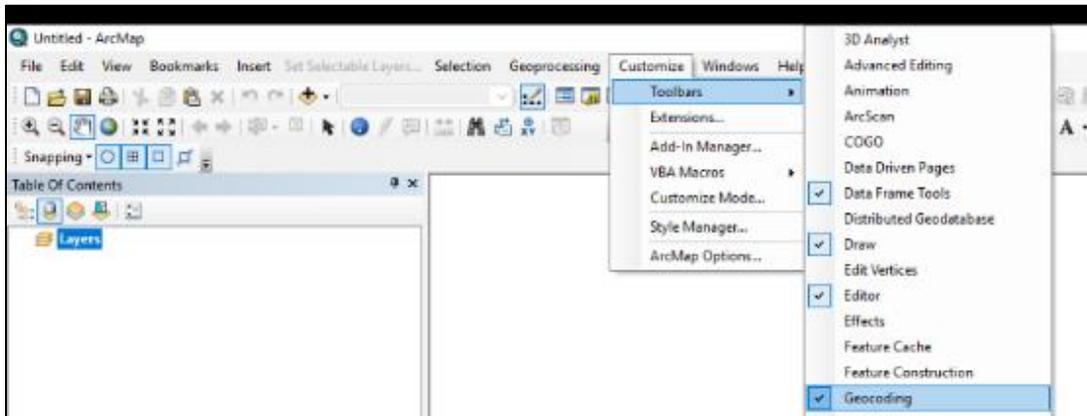
1. [ESRI ArcMap](#)
2. [ESRI ArcPro](#)
3. [QGIS](#)
4. [ESRI Rest Service](#)
 - i. [Geocoding a Single Address](#)
 - ii. [Reverse Geocoding](#)
 - iii. [Batch Geocoding](#)
 - iv. [Comparing Single and Batch Geocoding Operations](#)
5. [Python \(Jupyter\)](#)
 - i. [Single Address](#)
 - ii. [Batch Geocoding](#)
 - iii. [ArcGIS API for Python](#)
6. JavaScript/html
 - i. [Geocoding a Single Address](#)
 - ii. [Batch Geocoding](#)
7. [NYC Geographic Online Address Translator \(GOAT\)](#)

ESRI ArcMap

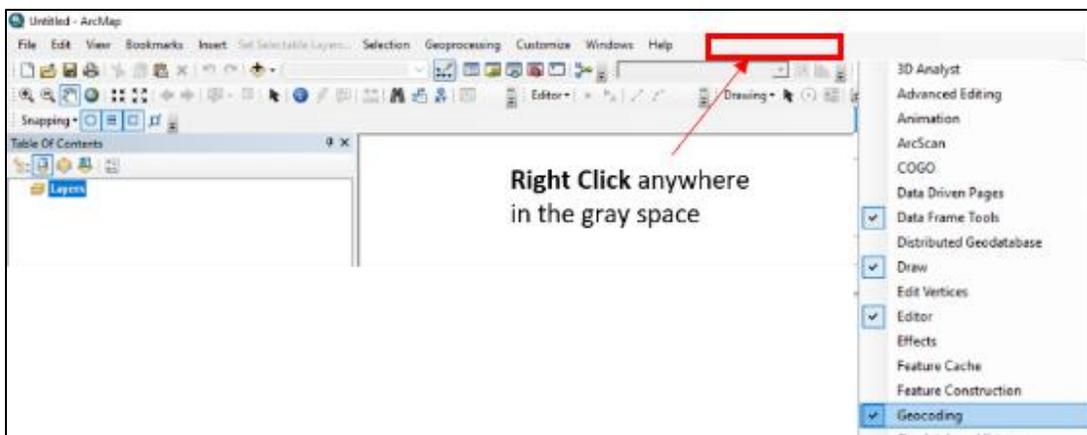
ESRI's ArcMap comes loaded with tools that allow the user to geocode addresses easily. Initially, the geocoding process requires two types of information, 1) reference data for creating an address locator and 2) input address data for matching. The reference data refers to a geographic information system (GIS) feature class containing the address attributes you want to search. For example, when searching for house number addresses, the reference data must contain the house number ranges, street names, city and/or zip code. For this purpose, we use the locators maintained by the SAM program area because they provide the most accurate dataset of addresses covering NYS.

Geocoding functionality is found in ArcMap's "Geocoding Toolbar." To ensure that this toolbar is activated either:

- Go to your Customize option in your top ArcMap toolbar and select Toolbars -> Geocoding to turn on the geocoding toolbar



- Or right click anywhere in the gray space along the top toolbar to bring up the selectable toolbars window



The Geocoding Toolbar looks like this:



The Toolbar should now be added to your ArcMap session

The Address Points maintained by the SAM team are the most accurate data available for NYS and should be picked anytime a result is returned from one of the SAM address point locators. If the geocoding service is used in the ESRI batch tool, the locator will generally return the “best address match” available from the GPO’s list of potential locators. It is important to note that the ESRI locators do not stop when a match is found, they continue to cascade through the rest of the locators within the composite, searching for the best overall match before returning a response. There are cutoffs established in the code where preferential or more accurate locators are weighted higher than others. For example, if an address is matched with a score of 100 against a street segment and a score of 95 against a rooftop point, the rooftop match will be returned as it is weighted higher due to its increased accuracy. The cutoff score for this type of weighting is set to 92, after which the street segment would be returned as the preferred match in this scenario. If there are multiple locators with the same score the first result is returned and it is coded as a Tie.

The locators will output a field named 'User_fld' which should be used in conjunction with the Loc_Name field. When the Loc_Name field contains one of the Address Point locators (1A, 1B or 1C) this field will contain either a 1,2,3,4 or a 5. When the Loc_Name field contains anything other than the Address Point locators, the 'User_fld' will either be NULL or "0". The numeric values correspond with the type of Address Point that was located:

1. Rooftop
2. Primary Structure Entrance
3. Driveway
4. Parcel Centroid
5. Miscellaneous

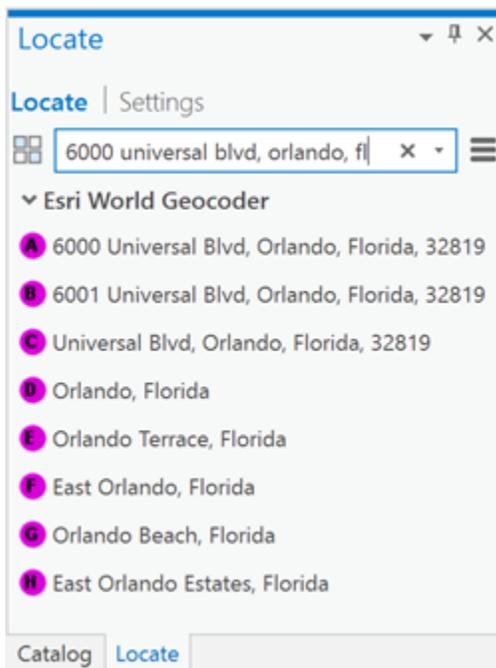
A pdf with detailed instructions on adding and using a locator in an ArcMap session can be found [Here!](#)

ESRI ArcPro

All information taken from ESRI online documentation [here](#).

Geocoding functions available in ArcGIS Pro

A great deal of geocoding functionality, both for beginners and for more advanced users who require more specialized geoprocessing tools, is available in the ArcGIS Pro application. You will find all of these geocoding functions in a few easy-to-find places. For example, the screen capture below shows the **Locate** pane, where you can quickly and easily geocode a single address.



The **Locate** pane is accessible only after you add a map to your application and you click the **Locate** button . If you want to geocode a single address, type it into the search box in the **Locate** pane pictured above, and press Enter. Your addresses display in list format shown below and in the image above as well as on the map. You can zoom to the addresses to determine which match you prefer.

The **Locate** pane will attempt to geocode any search input against all locators added to your project. Locators are automatically added to your project in ArcGIS Pro from your portal connection. If you are logged in with your ArcGIS Online account, the ArcGIS Online World Geocoding Service will be added automatically to your project, along with any other geocoding services available to you through your organizational account. Alternatively, if you are logged in through a custom portal, the geocoding services published on that portal as utility services will be added to your project.

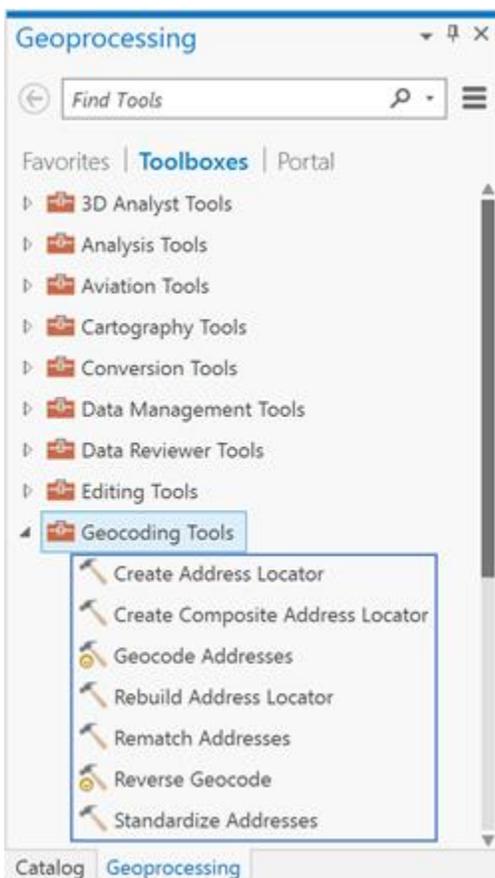
Note:

If you log out of your active portal and log in to a different portal in ArcGIS Pro, all portal locators will be refreshed, meaning locators from the initial portal will be removed and locators from the new portal will be added. This behavior applies only to locators that have been added as utility services and to the ArcGIS Online World Geocoding Service if you are signed in to ArcGIS Online. This behavior does not apply to locator items added manually to your project.

You can also [add additional locators to a project manually](#), from a file folder, from a server connection, or from locator items on your portal.

You can also modify some settings for geocoding from the **Settings** tab of the **Locate** pane. The **Settings** tab allows you to enable or disable locators in your project. If locators are enabled, they will be used to geocode addresses in the **Locate** pane and to reverse geocode if you right-click the map and select **What's here?**. If locators are not enabled, they will not be used for these two functions. Additionally, if a locator added to your project was built with suggestions capability, you can enable the use of suggestions for that locator in the **Locate** pane. By default, all locators are enabled for use in the **Locate** pane and with **What's here?**, and all locators with suggestions capabilities are used in the **Locate** pane.

The following screen capture shows another important location for geocoding related tools: the **Geoprocessing** pane, which allows you to access various geocoding tools.



To access the **Geoprocessing** pane:

1. Click the **Analysis** tab at the top of the page.
2. Click the **Tools** button .

The pane above appears on the side of your map. Use the Geocode Addresses tool to geocode multiple addresses at once. This process is explained more fully in [Geocode a table of addresses](#).

Note:

You can also geocode a table of addresses with the **Geocode Table** pane, which is accessed by adding a table to your project and then either right-clicking it and selecting **Geocode Table** or opening the **Data (Standalone Table)** tab of the ribbon and clicking the **Geocode Table** button . The **Geocode Table** pane will allow you to complete a guided workflow intended to streamline the process for beginning geocoding users or to complete the inputs of the pane manually. The **Geocode Table** tool will also offer improved performance in some instances.

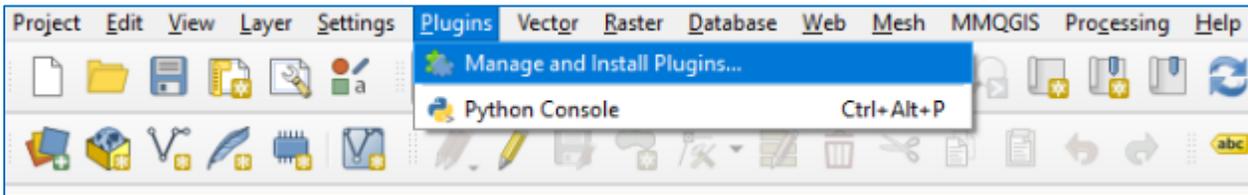
Extensive background GIS or geocoding knowledge is not necessary to use the geocoding functionality in ArcGIS Pro. You now have a basic understanding of the **Locate** pane, the **Geoprocessing** pane in the **Geocoding Tools** section, and the **Geocode Table** pane, which contain all the functionality you need to get started. Once you explore these three panes, every geocoding function available in ArcGIS Pro is ready for you to use.

QGIS

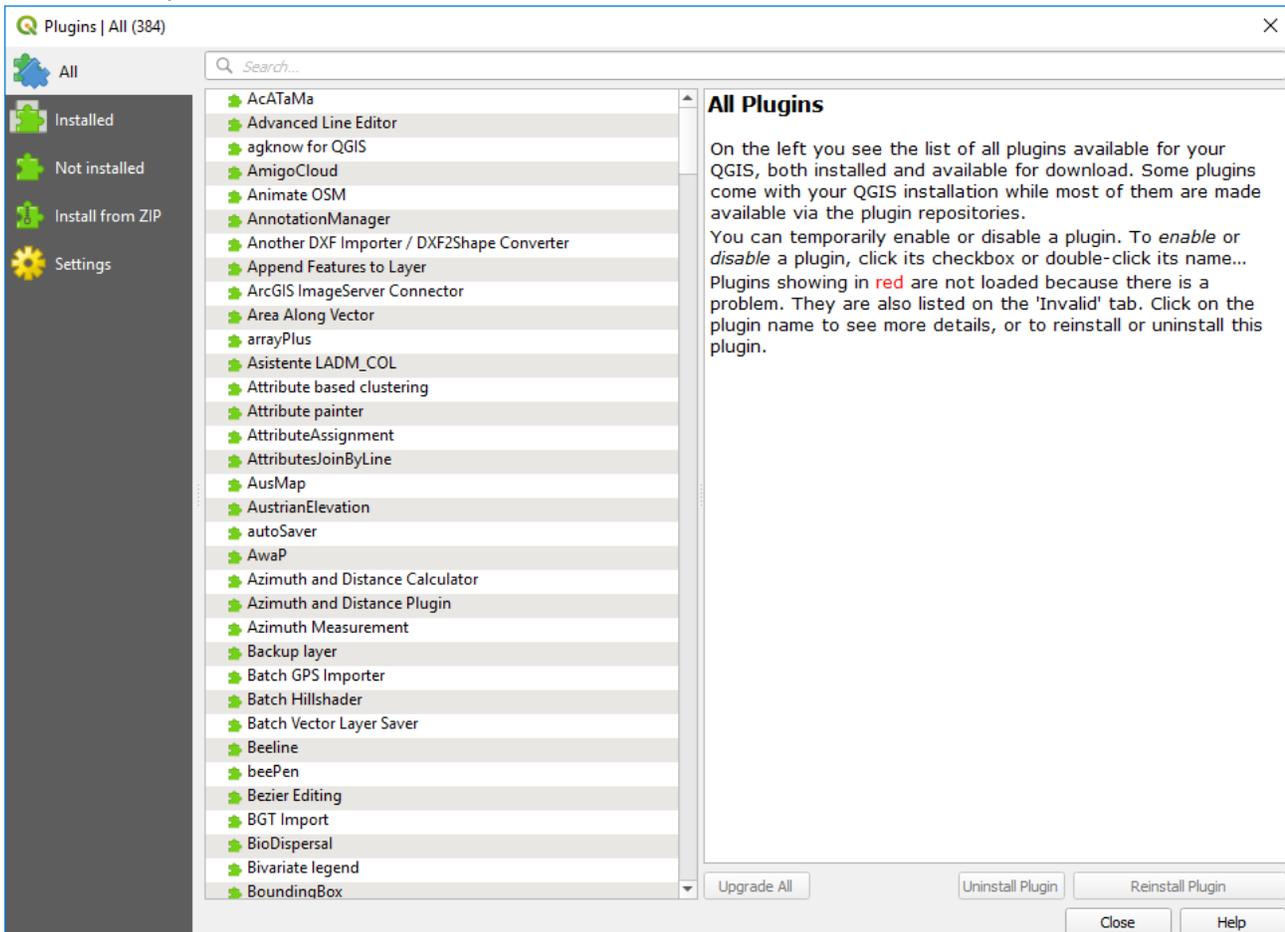
Recently, the MMQGIS geocoding plugin was updated by Professor Michael Minn of Farmingdale State College to allow for the utilization of ESRI based geocoding services.

Step1: Installing the MMQGIS Plugin

To install a QGIS plugin, click **Plugins** and then **Manage and Install Plugins...** from the top QGIS toolbar



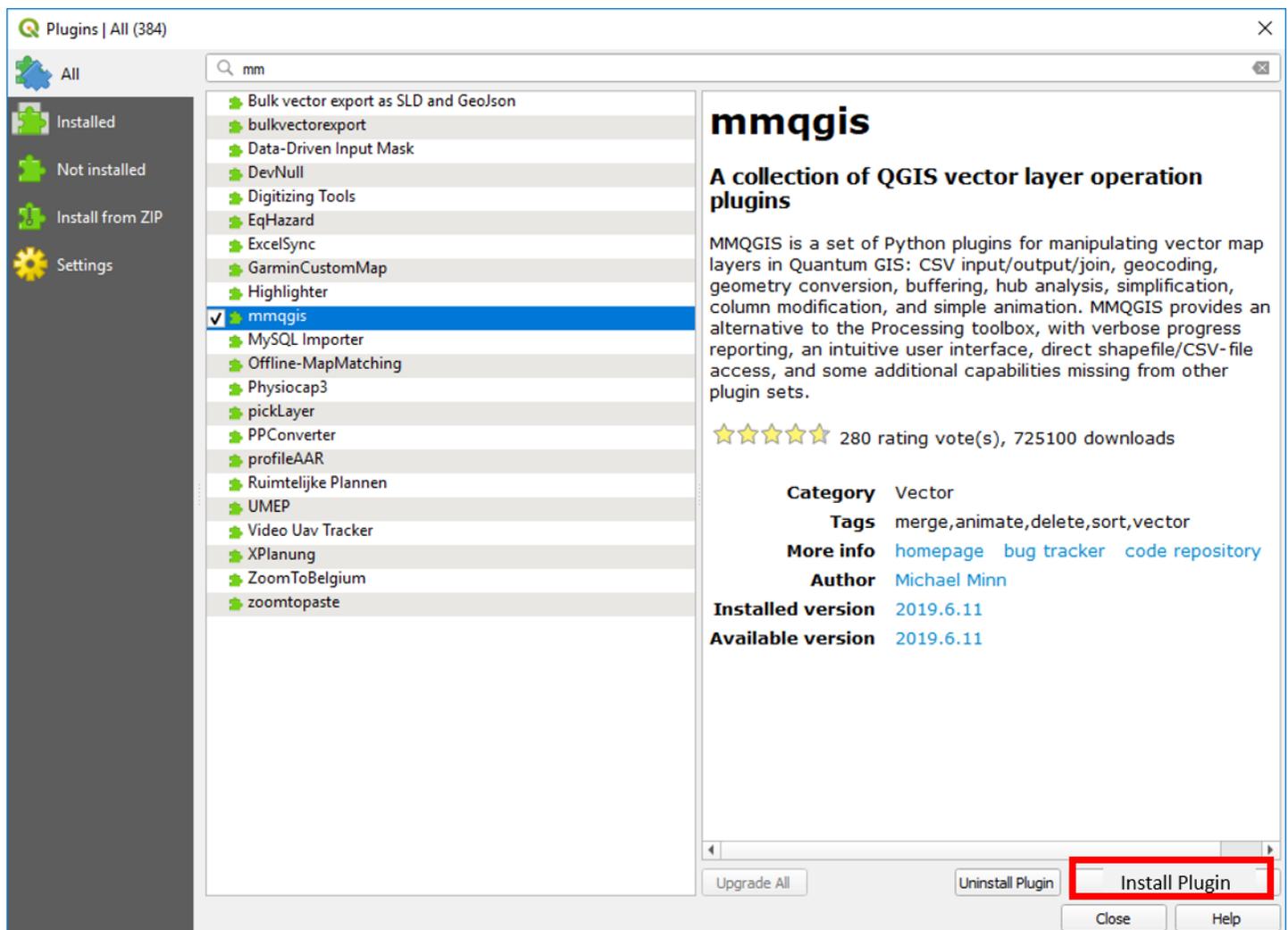
This will open a plugins window where you can add various plugins provided through the QGIS repository. You may be prompted that QGIS is “fetching repository information,” however just wait for the plugin window to open. It should look like this...



Notice that on the left-hand side of this window there are options to view plugins that are “Installed”, “Not Installed,” etc. For the purpose of this exercise, make sure that the option for “All” is selected. You should see an alphabetized list of plugins in the middle column.

Either in the search bar at the top of the window, or by scrolling down, locate the MMQGIS plugin.

Highlighting the MMQGIS plugin should change the dialogue on the right-hand side of the window to include a description of the MMQGIS plugin. In the lower right hand corner, you should also see a button labeled **Install Plugin**. Click the button and wait for the plugin to install (please note that if you have previously installed the MMQGIS plugin the button will say “Reinstall Plugin” which you should do in order to use the updated geocoding features. You are now ready to start geocoding.



Step 2: Geocoding with the MMQGIS plugin

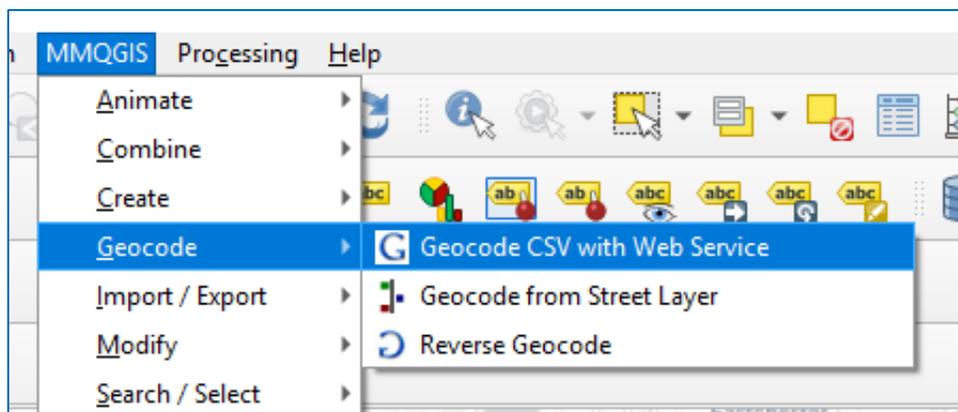
After installing the plugin, you should now see a MMQGIS option in the top QGIS toolbar.



Clicking MMQGIS will provide a dropdown list of supported activities; select Geocode to see the geocoding options available. The MMQGIS plugin provides three options for geocoding:

- 1) **Geocode CSV with Web Service** – Tool that imports addresses from a CSV file and uses a web geocoding service to geocode addresses to a point output file. The web service tools use the Python urllib module to make https requests to the respective geocoding APIs.
- 2) **Geocode Street Layer** – Tool that geocodes addresses from a CSV file using an address locator layer with street centerline features and attributes indicating the range of addresses associated with each feature.
- 3) **Reverse Geocode** – Tool that uses Google or Nominatim (OpenStreetMap) API to find addresses associated with point feature locations. If features are lines or polygons, the centroid of each feature will be used for querying.

Select the **Geocode CSV with Web Service** option.



This will open the **Web Service Geocode** window.

The inputs for this window are as follows:

- **Input CSV File:** CSV table of addresses to geocode. This file should be encoded in the UTF-8 character set. Although other 8-bit encodings (like Windoze ISO-8859-x) will work if only ASCII characters are present, non-ASCII characters may cause unpredictable behavior.
- **Address, City, State, and Country:** Selected columns from the input CSV file are added as attributes in the output shapefile. Addresses may be spread across as many as four different columns. However, these fields are concatenated into a single address to query the API, so only one meaningful column is absolutely required (such as for a city/state combination).

PLEASE NOTE: This means that even though the Address, City, State and Country dropdowns appear to require a like named corresponding column from your CSV file, **any CSV field may be entered into any of the dropdowns**. For example, a single address field may be entered into the State dropdown and all the other dropdown can be set to “none.” The plugin concatenates everything entered into all of the dropdowns and uses it to geocode.

- **Web Service:** The type of geocoder you wish to use. Choose from one of the four options provided: Google; OpenStreetMap/Nominatim; US Census Bureau; or ESRI Server. You will select the ESRI Server option.

- **API Key:** This field appears when either the Google, OpenStreetMap or US Census Bureau Geocoders are selected. To use these geocoders, you would need to get an API key and include it in the API Key dialog box.
- **ESRI Server URL:** When using an ESRI Server for geocoding, a server URL must be provided. Publicly accessible servers include:
 - ArcGIS World Geocoding Service:
<https://geocode.arcgis.com/arcgis/rest/services/World/GeocodeServer/findAddressCandidates>
 - NY State GIS Program Office Geocoding Service:
https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/findAddressCandidates

PLEASE NOTE: For the geocoding function to work, the url must be entered all the way down to the ESRI supported operation level. In the case of geocoding the url must therefore end at the “**findAddressCandidates**” operation.

- **Duplicate Handling:** This parameter indicates how to handle conditions where multiple results are returned for the same address.
 - **Use Only First Result:** As it suggests, this option retains only the first geocoded result. The result returned represents the best possible match determined by the geocoder after cascading through a series of locators.

PLEASE NOTE: The score for this match may not always be 100% even though it is the best match returned.

- **Multiple Features For Multiple Results:** This option retains all of the potential matches determined for a single address based on matches to different locators and the information stored within. Geocoders often return multiple locations representing different accuracy levels and spatial scales, for example: rooftop accuracy, street segment, parcel centroid, primary entrance, driveway, etc.). It is possible that since all these options exist inside the GPO’s locators you may see a match return for each. In the case of an apartment complex or other addressing anomaly, a large number of matches could be returned. Therefore, use of this option will likely require extensive editing of the results to cull unneeded points.
- **Output File Name:** The name of the output spatial / feature file that will be added to the QGIS map.
- **Not Found Output List:** The name of the output CSV file that will contain input addresses that could not be geocoded.

To begin geocoding:

- 1) Browse for and select an input CSV file to be geocoded.

Web Service Geocode

Input CSV File (UTF-8)
<file path>

Address: (none) City: (none)

State: (none) Country: (none)

Web Service: ESRI Server

ESRI Server URL

Duplicate Handling: Use Only First Result

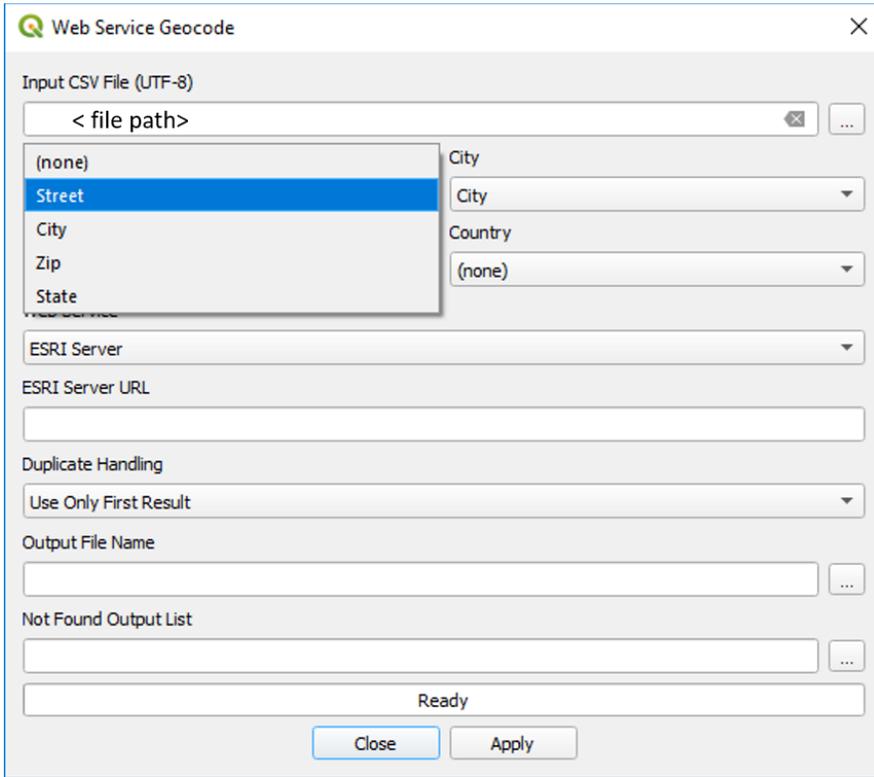
Output File Name

Not Found Output List

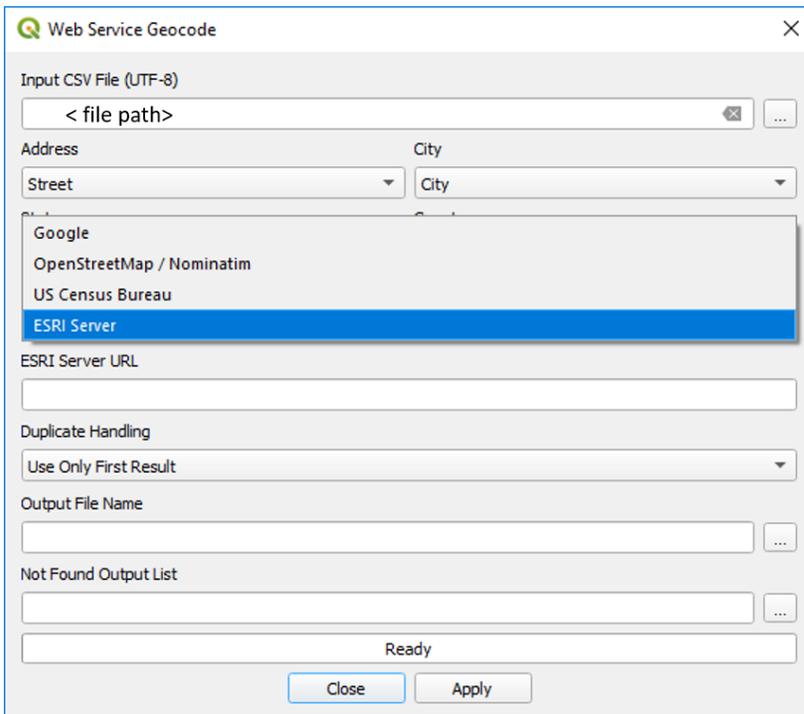
Ready

Close Apply

- 2) Determine which field in your input file will be used to geocode an address, for example Address, Street, SingleLine, or some combination of fields. Enter these fields into the MMQGIS window using the dropdown options, remembering that the dropdown name and your CSV field do not need to be the same.



3) Select the Web Service you want to use to geocode; in this case choose “ESRI Server”



- 4) Enter the ShareGIS url for the Streets and Addresses composite locator into the ESRI Server URL box

https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/findAddressCandidates

Web Service Geocode

Input CSV File (UTF-8)
< file path >

Address: Street
City: City
State: (none)
Country: (none)

Web Service: ESRI Server

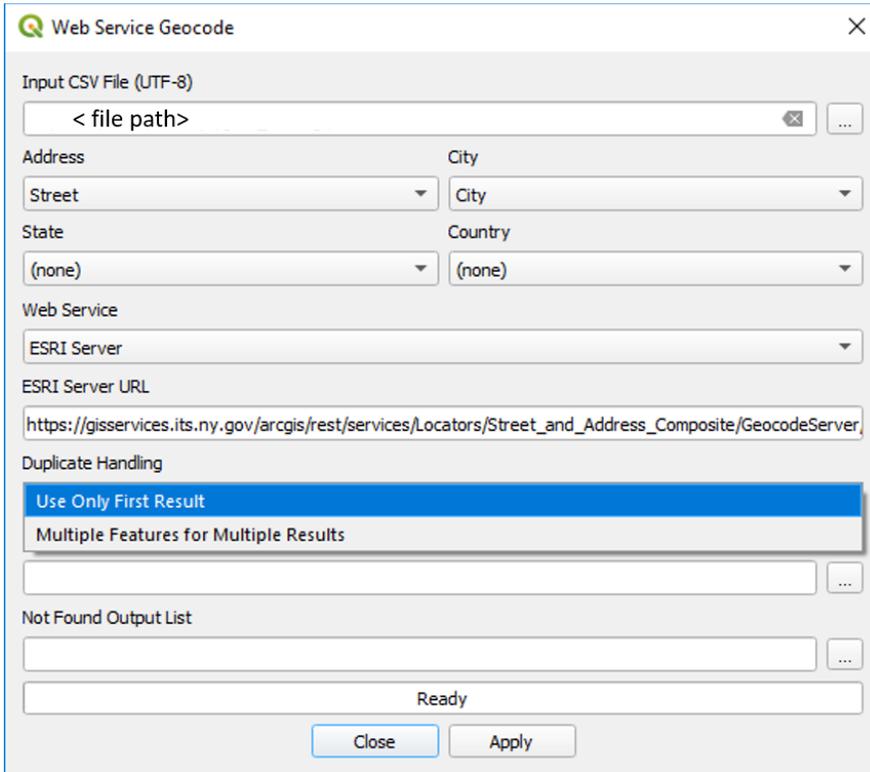
ESRI Server URL: https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer

Duplicate Handling: Use Only First Result

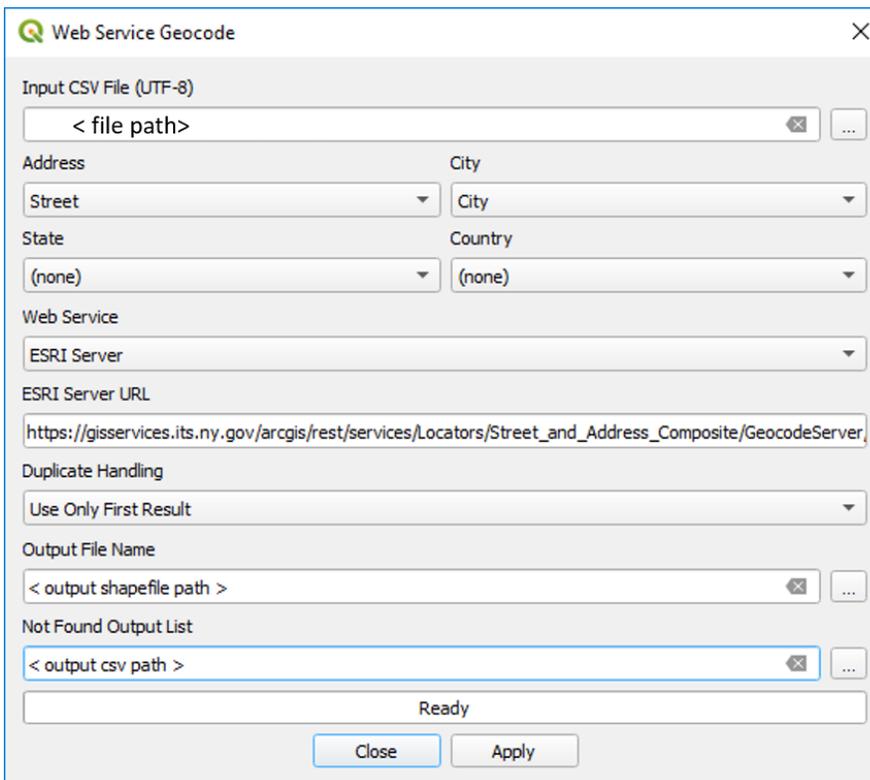
Output File Name:
Not Found Output List:
Ready

Close Apply

- 5) Determine if you want only the first result returned or multiple results per address.



6) Give a file name and output location for your spatial output file and not found list.



The output spatial file may take any of the following forms:

- ESRI Shapefile (*.shp)
- GEOJSON (*.geojson)
- KML (*.kml)
- Spatialite (*.sqlite)
- GPKG (*.gpkg)

The not found address file can take the form of either a .csv or .txt. file.

7) When you are ready to geocode click “**Apply**”

This will initiate the geocoding operation. The “Ready” bar visible on the dialogue box will begin to fill with the total number of successful geocodes completed until it runs through 100% of the input addresses. You may see a spinning icon for particularly large datasets, just wait for the operation to conclude.

Step 3: Understanding the Output

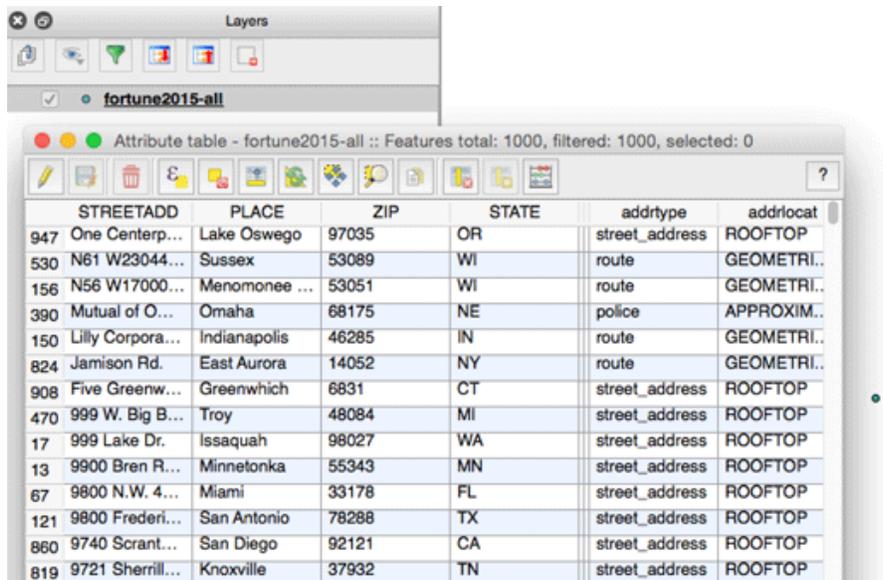
When viewing the output shapefile and csv from the MMQGIS geocoding operation it is important to know that all fields or attributes contained within the input file will become part of the output file. In addition, the following fields will be added to the output file:

FIELDS:

- result_num** = this is used if the “multiple features for multiple results” option is selected. The numbering is index origin or zero based numbering, so the first record will start with 0 and move up from there. The GPO geocoder cascades through the available locators with an algorithm, to determine the best match from all of the locators.
- score** = this is the score (0 – 100%) returned from the locator. The score may not be 100% even though the matching locator has been labeled as the best fit.
- address_ma** = this is the matching address that has been returned for the input address
- Loc_name** = this is the locator used to generate the match score
- Addr_type** = The type of address that was geocoded indicating what kind of feature the address was matched to
- User fld** = the point type, based off a set of number identifiers created by the streets and address program area. Possible results include:
- 0 = point is placed on the rooftop of the building
 - 1 = point is placed at the primary entry point for the address
 - 2 = point is placed at the driveway
 - 3 = point is placed at the parcel centroid
 - 4 = point is placed in a misc location
- latlong** = the return latitude and longitude coordinates generated for the address

Step 4. Viewing Your Results

The geocoded file will automatically be added to the layers list in QGIS. Once the geocoding is finished, the end result is a geocoded point shapefile. Appended to the table are two tables that describe how each point was determined.



The screenshot shows the QGIS interface. The Layers panel at the top left displays a layer named 'fortune2015-all'. Below it, the attribute table for this layer is open, showing 1000 features. The table has columns for STREETADD, PLACE, ZIP, STATE, addrtype, and addrlocat. The first few rows of data are as follows:

	STREETADD	PLACE	ZIP	STATE	addrtype	addrlocat
947	One Centerp...	Lake Oswego	97035	OR	street_address	ROOFTOP
530	N61 W23044...	Sussex	53089	WI	route	GEOMETRI..
156	N56 W17000...	Menomonee ...	53051	WI	route	GEOMETRI..
390	Mutual of O...	Omaha	68175	NE	police	APPROXIM..
150	Lilly Corpora...	Indianapolis	46285	IN	route	GEOMETRI..
824	Jamison Rd.	East Aurora	14052	NY	route	GEOMETRI..
908	Five Greenw...	Greenwich	6831	CT	street_address	ROOFTOP
470	999 W. Big B...	Troy	48084	MI	street_address	ROOFTOP
17	999 Lake Dr.	Issaquah	98027	WA	street_address	ROOFTOP
13	9900 Bren R...	Minnnetonka	55343	MN	street_address	ROOFTOP
67	9800 N.W. 4...	Miami	33178	FL	street_address	ROOFTOP
121	9800 Frederi...	San Antonio	78288	TX	street_address	ROOFTOP
860	9740 Scrant...	San Diego	92121	CA	street_address	ROOFTOP
819	9721 Sherrill...	Knoxville	37932	TN	street_address	ROOFTOP

To the right of the attribute table, a map view shows a cluster of green circular points representing the geocoded locations.

ESRI Rest Service

The ShareGIS services environment uses an ESRI ArcServer framework which dictates the use of ESRI rest endpoints for connecting to web services. All of the geocoding webservice are accessed via <https://gisservices.its.ny.gov/arcgis/rest/services/locators>. Each of the web services associated with the composite locators mentioned [above](#) provides a user interface for submitting an address to be geocoded. The user interface for each of the composite locators provides three functions for the user when accessed via rest service; these options are:

- Find Address Candidates = geocoding a single address
- Reverse Geocode = finding an address based off a coordinate pair
- Geocode Addresses = batch geocoding option provided with ESRI web services

The locators are all available in the [service catalogue](#) and are replicated in both ShareGIS environments:

[Non-Production](#) and [Production](#)

Before Geocoding:

- Evaluate which locator is the best fit for the address information you have.
- Determine if the user is attempting a Single or Batch geocoding operation
- Determine if the user has any coding experience as there are certain limitations associated with using a rest service, such as:
 - The url interface does not allow for the uploading of a csv file or table; addresses need to be entered manually in a specific format
 - There may be a limit to the total number of addresses that can be entered for a batch geocoding operation (still needs to be verified)

Geocoding a Single Address Via Rest Service

NOTE: geocoding a single address as a batch with the “Geocode Addresses” functionality provides more information than performing a single geocode with the “findAddressCandidates” functionality

Steps

1. First determine the type of address information the user has:
 - Complete address containing Number, Street, City, State and Zip Code – use the Streets and Address Composite locator
 - Address missing Number or Zip Code – use the Street NoNum and ZipCode Composite locator
 - Only a place name – use the NYPlace locator
2. Once the type of address information is determined select the appropriate locator url.

- Go to https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer
- At the bottom of the webpage select “Find Address Candidates”

Locator Properties:

- UICLSID: {3D486637-6BCF-4A0C-83DB-A02D437FB8FC}
- SuggestedBatchSize: 1000
- MaxBatchSize: 1000
- LoadBalancerTimeOut: 60
- WriteXYCoordFields: FALSE
- WriteStandardizedAddressField: FALSE
- WriteReferenceIDField: FALSE
- WritePercentAlongField: FALSE

Supported Operations: [Find Address Candidates](#) [Reverse Geocode](#) [Geocode Addresses](#)

Child Resources: [Info](#)

- The NYS geocoding service and all of the locators use the “findAddressCandidates” function to geocode a single address, and the Find Address Candidates window is the same for all locators. With this functionality, you have the option of using either a:

Multifield Address Approach = enter in each of the address’ unique parts

Or

Single Line Approach = enter the address on one line separated by commas

PLEASE NOTE: One of the know limitations of the address locators revolves around the inclusion of the State field. In some cases, the State field is causing a high score for an address point in New York City where not appropriate. An example of this is if you enter 620 Madison St, Syracuse NY 13210 the geocoder returns 620 Madison St, New York, NY 11221. To avoid this error when geocoding with the Street_and_Address_Composite, the GIS Program Office recommends not mapping the State field. If you enter 620 Madison St, Syracuse 13210, the correct location is

returned. The State field can still be mapped in the Street_NoNum_and_ZipCode_Composite as there are no longer any address point locators within this composite to cause this issue.

6. The remaining fields can be left blank. However, it is advised to enter an Output Spatial Reference. The default Output Spatial Reference is NAD 1983 if nothing is entered, however the output for NAD 1983 is in decimal meters that may not be easy for the user to understand. Using 4326 (Web Mercator) provides a more understandable longitude, latitude format in decimal degrees for the result.

NAD 1983

vs.

WGS 84

```
Shape:
Point:
X: 601644.6003526772
Y: 4723145.3878174145
Score: 95.0
Address: 80 S Swan St, Albany, NY, 12210
```

```
Shape:
Point:
X: -73.75989873712614
Y: 42.654069473771784
Score: 95.0
Address: 80 S Swan St, Albany, NY, 12210
```

Additionally, users can specify which output fields will be returned (entering "*" will return all fields) and select the desired format from the "Format" dropdown menu (choose from HTML, JSON or KMZ). Once the address is entered select "Find Address Candidates" at the bottom of the screen.

Output example for:

80 South Swan Street, Albany, 12210

The screenshot shows a web-based address search tool. The input fields include: Street (empty), City (empty), State (empty), ZIP (empty), SingleLine (80 South Swan Street, Alb), Category (empty), Out Fields (*), Max Locations (empty), Output Spatial Reference (4326), Search Extent (empty), Location (empty), Distance in Meters (empty), Magic Key (empty), and Format (HTML). Below the input fields are two buttons: "Find Address Candidates (GET)" and "Find Address Candidates (POST)".

The output section is titled "Address Candidates: (# address candidates : 3)". It displays the following information:

```
Shape:
Point:
X: -73.75989873712614
Y: 42.654069473771784
Score: 95.0
Address: 80 S Swan St, Albany, NY, 12210
Loc_name: 1A_SAM_AP_ZipN
Score: 95
Match_addr: 80 S Swan St, Albany, NY, 12210
NumberPrefix:
House: 80
NumberSuffix:
```

It is common for more than one address candidate result to be returned. When searching for "candidates" the tool returns any possible candidate that comes back above the score threshold for that address. It is therefore possible to get more than one candidate from the same locator. For example, if there is an alternate street name for the given address, it would be returned as an

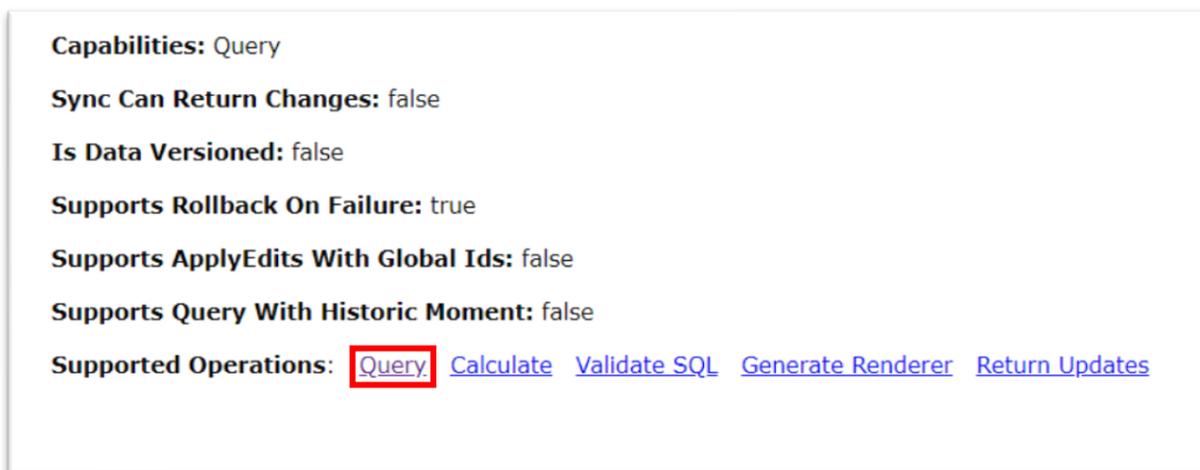
alternative candidate. It is also possible to get an alternate candidate from one of the other Address Point or Street locators. For example, if the City name is used as input, it may also find the address point by using the ZipCode. The coordinates for each of the returned matches will likely vary based on if the match is from a point or street locator. It is possible to get the same score, but the locations will be in different places since the street locators are interpolated along the street range.

Note: Which output spatial reference you used is important to remember

The power of the NYS Geocoder and the additional web services provided by ShareGIS is that you can use the output coordinates of the geocoder to find the attributes of another service for that location. For example, if you wanted to know what county your address is in you could follow these steps:

Steps

1. Go to the ShareGIS web service for NYS Counties
2. [At the bottom of this page](#) select “Query”



3. In the resulting query window fill out the fields as follows to return a result:
 - Enter the coordinates from your geocoding result in the format of X coordinate, Y coordinate, remembering to add the negative sign in front of the X coordinate.
 - Geometry Type = “Point”
 - Input Spatial Reference = Whichever Spatial Reference you used to generate the address coordinates, for example 4326
 - Spatial Relationship = “Intersects”
 - Format = The output format can be either 1) HTML; 2) JSON; or 3) KMZ (the zipped format for KML)
 - Leave all remaining fields as the default
4. The county your address falls in will be displayed at the bottom

Output for 80 South Swan Street, Albany, 12210

Where:

Text:

Object IDs:

Time:

Input Geometry:

Geometry Type:

Input Spatial Reference:

Spatial Relationship:

Relation:

Out Fields:

Return Geometry: True False

Return True Curves: True False

Max Allowable Offset:

Geometry Precision:

Output Spatial Reference:

Return IDs Only: True False

Return Count Only: True False

Order By Fields:

Group By Fields (For Statistics):

Output Statistics:

ReturnZ: True False

ReturnM: True False

Geodatabase Version Name:

Return Distinct Values: True False

Result Offset:

Result Record Count:

Query By Distance:

Return Extents Only: True False

Datum Transformation:

Parameter Values:

Range Values:

Format:

Query (GET) Query (POST)

x,y for 80 South Swan St.

This is the Spatial Reference you obtained your coordinates in when geocoding

records: 1

NAME: Albany
Polygon:

This is the County your address falls in

[-8201679.638599999, 5279062.881099999] , [-8201717.1281, 5278892.5757] , [-8201850.206499999, 5278389.238200001] more...

Reverse Geocoding

ESRI locator services use the “Reverse Geocode” function to do just that.

Steps

1. Go to https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer
2. At the bottom of the webpage select “Reverse Geocode”

Locator Properties:

- UICLSID: {3D486637-6BCF-4A0C-83DB-A02D437FB8FC}
- SuggestedBatchSize: 1000
- MaxBatchSize: 1000
- LoadBalancerTimeOut: 60
- WriteXYCoordFields: FALSE
- WriteStandardizedAddressField: FALSE
- WriteReferenceIDField: FALSE
- WritePercentAlongField: FALSE

Supported Operations: [Find Address Candidates](#) [Reverse Geocode](#) [Geocode Addresses](#)

Child Resources: [Info](#)

3. On the resulting screen enter your x, y coordinate pair in the “Location input” box. The structure of the coordinate pair should be entered in either JSON point structure, or you can specify the location with a simple comma-separated syntax.

Syntax Examples for 80 South Swan Street, Albany, NY, 12210:

- JSON object

```
{"x" : -73.75989873712614, "y" : 42.654069473771784, "spatialReference" : {"wkid" : 4326}}
```
- Simple syntax

```
{"x" : 601644.6003526772, "y": 4723145.3878174145}
```

The Spatial Reference is declared as part of the input where the syntax reads
“spatialReference” : {“wkid” : 4326}

This informs the reverse geocoding function which spatial reference that the coordinates are in. **If the spatial reference or wkid is not specified in the JSON object, the reverse geocoding function assumes you are using the default spatial reference of NAD 83 and it will not accept coordinates in a different format.** If you're using the simple comma-separated syntax, the location is assumed

to be in the same spatial reference as that of the geocode service which is NAD 83. This is why two separate coordinate formats are listed in the syntax example above.

JSON Object

Location:	<input "="" type="text" value='{"x": -73.75989873712614, "y": 42.654069473771784, "spatialReference": {"wkid": 426983}}'/>
Distance:	<input type="text"/>
Language Code:	<input type="text"/>
Output Spatial Reference:	<input type="text"/>
Return Intersection:	<input type="radio"/> True <input checked="" type="radio"/> False
Format:	<input type="text" value="HTML"/>
<input type="button" value="Reverse Geocode (GET)"/> <input type="button" value="Reverse Geocode (POST)"/>	

Reverse Geocoded Address:

Street: 80 S Swan St
City: Albany
State: NY
ZIP: 12210
Loc_name: 1A_SAM_AP_ZipN

Reverse Geocoded Location:

Point:
X: 601644.6003526772
Y: 4723145.3878174145
Spatial Reference: 26918 (26918)

Simple Syntax

Location:	<input "="" type="text" value='{"x": 601644.6003526772, "y": 4723145.3878174145}'/>
Distance:	<input type="text"/>
Language Code:	<input type="text"/>
Output Spatial Reference:	<input type="text"/>
Return Intersection:	<input type="radio"/> True <input checked="" type="radio"/> False
Format:	<input type="text" value="HTML"/>
<input type="button" value="Reverse Geocode (GET)"/> <input type="button" value="Reverse Geocode (POST)"/>	

Reverse Geocoded Address:

Street: 80 S Swan St
City: Albany
State: NY
ZIP: 12210
Loc_name: 1A_SAM_AP_ZipN

Reverse Geocoded Location:

Point:
X: 601644.6003526772
Y: 4723145.3878174145
Spatial Reference: 26918 (26918)

Batch Geocoding Against the ESRI Service

Similar to geocoding a single address, you can batch geocode multiple addresses at a time using the “Geocode Addresses” functionality of Arc Server. **Batch geocoding functionality off the rest service does not allow a user to upload a file of addresses for a batch process. Instead addresses must be manually entered on the “Geocode Addresses” screen.**

Steps

1. Go to
https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer
2. At the bottom of the webpage select “Geocode Addresses”



Locator Properties:

- UICLSID: {3D486637-6BCF-4A0C-83DB-A02D437FB8FC}
- SuggestedBatchSize: 1000
- MaxBatchSize: 1000
- LoadBalancerTimeOut: 60
- WriteXYCoordFields: FALSE
- WriteStandardizedAddressField: FALSE
- WriteReferenceIDField: FALSE
- WritePercentAlongField: FALSE

Supported Operations: [Find Address Candidates](#) [Reverse Geocode](#) [Geocode Addresses](#)

Child Resources: [Info](#)

3. On the resulting screen, in the addresses text box, enter as many addresses as you like using the following format:

```
{"records": [ { "attributes": { "OBJECTID": 1, "SINGLELINE": "ADDRESS 1" } }, { "attributes": { "OBJECTID": 2, "SINGLELINE": "ADDRESS 2" } }, { "attributes": { "OBJECTID": 3, "SINGLELINE": "ADDRESS 3" } }, { "attributes": { "OBJECTID": 4, "SINGLELINE": "ADDRESS 4" } } ] }
```

With a new “OBJECTID” and “ADDRESS” entry for each address to be geocoded.

4. Enter 4326 for the Output Spatial Reference (or the spatial reference of your choice)
5. At the bottom of the page select “Geocode Addresses (GET)”

Geocode Addresses: (Locators/Street_and_Address_Composite)

Addresses:

Category:

Source Country:

Output Spatial Reference:

Format:

Geocode Addresses (GET) Geocode Addresses (POST)

Geocoded Addresses:

address: 10 B Airline Dr, Albany, NY, 12205
location: {"x":-73.81704519699997,"y":42.73466108300005}
score: 95
attributes: {"ResultID":1,"Loc_name":"1A_SAM_AP_ZipN","Status":"M","Score":95,"Match_addr":"10 B Airline Dr, Albany, NY, 12205","Side":"","SubAdd":"","User_fld":"1","Addr_type":"StreetAddress"}

address: 80 S Swan St, Albany, NY, 12210
location: {"x":-73.75989873699996,"y":42.65406947400004}
score: 100
attributes: {"ResultID":2,"Loc_name":"1A_SAM_AP_ZipN","Status":"M","Score":100,"Match_addr":"80 S Swan St, Albany, NY, 12210","Side":"","SubAdd":"","User_fld":"1","Addr_type":"StreetAddress"}

address: 40 Wolf Rd, Albany, NY, 12205
location: {"x":-73.81774103599997,"y":42.71235314000006}
score: 95
attributes: {"ResultID":1,"Loc_name":"1A_SAM_AP_ZipN","Status":"M","Score":95,"Match_addr":"40 Wolf Rd, Albany, NY, 12205","Side":"","SubAdd":"","User_fld":"1","Addr_type":"StreetAddress"}

Comparing ESRI Single and Batch Geocoding Operations

It is important to note that when the “findAddressCandidates” (single) and the “Geocode Addresses” (batch) functions were first designed, they were designed with differing logic components. Therefore, it is likely the “findAddressCandidates” and “Geocode Addresses” functions will generate different results for the same address.

“findAddressCandidates” for 80 South Swan Street, Albany, NY, 12210

```
Address Candidates: (# address candidates : 3)  
  
Shape:  
  
    Point:  
    X: -73.75989873712614  
    Y: 42.654069473771784  
  
Score: 95.0  
Address: 80 S Swan St, Albany, NY, 12210  
Loc_name: 1A_SAM_AP_ZipN  
Score: 95  
Match_addr: 80 S Swan St, Albany, NY, 12210  
NumberPrefix:  
House: 80  
NumberSuffix:  
Side:  
PreModifier:  
PreDir: S  
PreType:  
SeparatorElement:  
StreetName: Swan  
SufType: St  
SufDir:  
PostModifier:  
SubAdd:  
City: Albany  
State: NY  
ZIP: 12210  
User_fld: 1  
Addr_type: StreetAddress  
FromAddr:  
ToAddr:
```

vs.

“Geocode Addresses” for 80 South Swan Street, Albany, NY, 12210

```
Geocoded Addresses:  
  
address: 80 S Swan St, Albany, NY, 12210  
location: {"x":-73.75989873699996,"y":42.65406947400004}  
score: 95  
attributes: {"ResultID":1,"Loc_name":"1A_SAM_AP_ZipN","Status":"M","Score":95,"Match_addr":"80 S Swan St, Albany, NY,  
12210","Side":"","SubAdd":"","User_fld":"1","Addr_type":"StreetAddress"}
```

Geocoding with Python (Jupyter Notebook against the service)

The Jupyter notebook provides a convenient way of working with python code, however the following code can be used in any python editor.

The necessary libraries you will need to run this code in python are “requests” and “json”

Geocoding a Single Address (Python)

Geocoding with python requires formatting a url against the geocoding service that will be used to generate the desired outcome.

Steps

1. URL Format

The url for generating a json formatted response when geocoding 10B Airline Drive, Albany, NY 12235 in output spatial reference 4326 is as follows...

```
'https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/findAddressCandidates?Street=&City=&State=&ZIP=&SingleLine=80+South+Swan+Street%2C+Albany%2C+NY%2C+12210&category=&outFields=&maxLocations=&outSR=4326&searchExtent=&location=&distance=&magicKey=&f=json'
```

This url has several parts:

- The domain and path of the geocoding service
https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer
- The specific function we are looking to use, in this case we want to geocode a single address, so we use the `findAddressCandidates` function
- `?` designates the start of the query against the service
- All of the potential input fields are listed out in the url, and if they are blank the format follows a pattern of field=(blank), so in the url above we see each field followed by an equal sign (=) and an ampersand (&) e.g., “`Street=&City=&State=&ZIP=&`” This format shows that each of the fields (Street, City, State and ZIP) have no designated value with the ampersand adding on the next field contained (but not required to have a value) in the function.
- When a field does have a value it is entered in the url as a value after the equals sign (=). In the above url we have a value for the SingleLine field,
`SingleLine=80+South+Swan+Street%2C+Albany%2C+NY%2C+12210`

With “+” representing a space and “%2C” representing a “,” in the address.

- After all of the fields are entered, the “f=” represents the requested format. In the case of this example the requested return format is Json, so either “f=json” or “f=pjson” can be used.
 - i. “f=json” is the preferred format for production applications as a degradation in performance may be experienced while using pjson. The output format will look similar to this (image cropped):

```
{
  "results": [
    {
      "layerId": 0,
      "layerName": "NYS_Senate_Districts",
      "displayFieldName": "NAME",
      "value": "Catharine Young",
      "attributes": {
        "OBJECTID_1": "53",
        "NAME": "Catharine Young",
        "TOTAL_ADJ": "292081",
        "POPULATION": "292369",
        "IDEAL_VALU": "307356",
        "DEVIATION": "-15275",
        "F_DEVIATIO": "-0.0497",
        "F_POPULATI": "1.001",
        "PARTY": "Republican",
        "PHONE": "518-455-3563",
        "EMAIL": "cyoung@senate.state.ny.us",
        "DIST_PHONE": "716-372-4901",
        "SHAPE_LENG": "7.35968",
        "DISTRICT": "57",
        "SHAPE": "Polygon",
        "SHAPE.AREA": "10770185310.314199",
        "SHAPE.LEN": "680914.600139"
      },
      "geometryType": "esriGeometryPolygon",
      "geometry": {
        "rings": [
          [
            -77.749933729110595,
            41.9987903587148
          ]
        ]
      }
    }
  ]
}
```

- ii. “f=pjson” provides a more readable format for the json response, adding padding, and is preferable for development or debugging. The output format will look like this:

```
{
  "results": [
    {
      "layerId": 0,
      "layerName": "NYS_Senate_Districts",
      "displayFieldName": "NAME",
      "value": "Catharine Young",
      "attributes": {
        "OBJECTID_1": "53",
        "NAME": "Catharine Young",
        "TOTAL_ADJ": "292081",
        "POPULATION": "292369",
        "IDEAL_VALU": "307356",
        "DEVIATION": "-15275",
        "F_DEVIATIO": "-0.0497",
        "F_POPULATI": "1.001",
        "PARTY": "Republican",
        "PHONE": "518-455-3563",
        "EMAIL": "cyoung@senate.state.ny.us",
        "DIST_PHONE": "716-372-4901",
        "SHAPE_LENG": "7.35968",
        "DISTRICT": "57",
        "SHAPE": "Polygon",
        "SHAPE.AREA": "10770185310.314199",
        "SHAPE.LEN": "680914.600139"
      },
      "geometryType": "esriGeometryPolygon",
      "geometry": {
        "rings": [
          [
            -77.749933729110595,
            41.9987903587148
          ]
        ]
      }
    }
  ]
}
```

2. The primary issue with these types of requests to a service url is that a url “get request” response comes back as a string when using `requests.get()` and does not return in a python dictionary. Therefore, we are unable to see the properly formatted json even though the response format is requested to be json. So a transformation is first needed to take the result of the get request and transform into a python dictionary. This is done with the code `json.loads(requests.get(url).text)`.

Sample Code

```
In [1]: import requests

import json

In [2]: url = 'https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/findAddressCandidates?Street=&City=&State=&ZIP=&SingleLine=10+B+Airline+Drive%2C+Albany%2C+NY%2C+12235&category=&outFields=&maxLocations=&outSR=4326&searchExtent=&location=&distance=&magicKey=&f=pjson'

In [3]: ResponseDictionary = json.loads(requests.get(url).text)

In [4]: ResponseDictionary
Out [4]: {'candidates': [{'address': '10 B Airline Dr, Albany, NY, 12205',
'attributes': {},
'location': {'x': -73.81704519669566, 'y': 42.734661082568536},
'score': 95},
{'address': '10 Airline Dr, Albany, NY, 12205',
'attributes': {},
'location': {'x': -73.81864764620448, 'y': 42.737130745418085},
'score': 95}],
'spatialReference': {'latestWkid': 4326, 'wkid': 4326}}
```

Notice that two potential address candidates are returned.

[Batch Geocoding Addresses \(Jupyter Notebook\)](#)

Steps

1. Batch geocoding in python is similar to geocoding a single address. The process requires the same libraries and begins with formatting a url.

The url for generating a json formatted response when geocoding three addresses (10B Airline Drive, Albany, NY 12235; 800 North Pearl Street, Albany, NY, 12204; and 80 South Swan Street, Albany, NY12210) in output spatial reference 4326 is as follows...

```
'https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/geocodeAddresses?addresses=%7B%22records%22%3A+%5B+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+1%2C+%22SINGLELINE%22%3A+%2210B+Airline+Drive%2C+Albany%2C+NY%2C+12235%22%7D+%7D%2C+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+2%2C+%22SINGLELINE%22%3A+%22800+N+Pearl+St%2C+Menands%2C+N
```

Y+12204%22%7D+%7D+%2C+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+1%2C+%22SINGLELINE%22%3A+%2280+S+Swan+St%2C+Albany%2C+NY+12210%22%7D%7D%5D%7D&category=&sourceCountry=&outSR=4326&f=json

Again, this url has several parts:

- The domain and path of the geocoding service
https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer
- The specific function we are looking to use, in this case we want to geocode multiple addresses, so we use the **geocodeAddresses** function
- **?** designates the start of the query against the service
- Similar to the **“findAddressCandidates”** function the **“geocodeAddresses”** function starts off with all of the potential input fields listed out in the url, and if they are blank the format follows a pattern of field=(blank), so in the url above we see each field followed by an equal sign (=) and an ampersand (&) e.g., **“Street=&City=&State=&ZIP=&”** This format shows that each of the fields (Street, City, State and ZIP) have no designated value with the ampersand adding on the next field contained (but not required to have a value) in the function.
- When a field does have a value it is entered in the url as a value after the equals sign (=). In the above url we have a value for each of the three addresses we are attempting to geocode inside of the SingleLine field,S
 - i. **22SINGLELINE%22%3A+%2210B+Airline+Drive%2C+Albany%2C+NY%2C+12235%22%7D+%7D%2C+%7B**
 - ii. **SINGLELINE%22%3A+%22800+N+Pearl+St%2C+Menands%2C+NY+12204%22%7D+%7D+%2C+%7B**
 - iii. **SINGLELINE%22%3A+%2280+S+Swan+St%2C+Albany%2C+NY+12210%22%7D%7D%5D%7D**
- Additional characters are used in this code:
 - i. **+** = a space
 - ii. **%2C** = ,
 - iii. **%22** = “
 - iv. **%3A** = :
 - v. **%5D** =]
 - vi. **%5B** = [
 - vii. **%7D** = }
 - viii. **%7B** = {
- After all of the fields are entered, the **“f=”** represents the requested format. In the case of this example the requested return format is padded json, so **“f=json”** is used.

2. Results from the service come back as a string when using `requests.get()`; therefore it needs to be converted into a python dictionary.

Sample Code

```
In [1]: import requests

import json
```

```
In [2]: url = 'https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer/geocodeAddresses?addresses=%7B%22records%22%3A+%5B+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+1%2C+%22SINGLELINE%22%3A+%2210B+Airline+Drive%2C+Albany%2C+NY%2C+12235%22%7D+%7D%2C+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+2%2C+%22SINGLELINE%22%3A+%22800+N+Pearl+St%2C+Menands%2C+NY+12204%22%7D+%7D%2C+%7B+%22attributes%22%3A+%7B+%22OBJECTID%22%3A+1%2C+%22SINGLELINE%22%3A+%2280+S+Swan+St%2C+Albany%2C+NY+12210%22%7D%7D%5D%7D&category=&sourceCountry=&outSR=4326&f=json'
```

```
In [3]: ResponseDictionary = json.loads(requests.get(url).text)
```

```
In [4]: ResponseDictionary
```

```
Out [4]: {'locations': [{'address': '10 B Airline Dr, Albany, NY, 12205',
  'attributes': {'Addr_type': 'StreetAddress',
  'Loc_name': '1A_SAM_AP_ZipN',
  'Match_addr': '10 B Airline Dr, Albany, NY, 12205',
  'ResultID': 1,
  'Score': 95,
  'Side': '',
  'Status': 'M',
  'SubAdd': '',
  'User_fld': '1'},
  'location': {'x': -73.81704519699997, 'y': 42.73466108300005},
  'score': 95},
 {'address': '800 N Pearl St, Albany, NY, 12204',
  'attributes': {'Addr_type': 'StreetAddress',
  'Loc_name': '1A_SAM_AP_ZipN',
  'Match_addr': '800 N Pearl St, Albany, NY, 12204',
  'ResultID': 2,
  'Score': 96.46,
  'Side': '',
  'Status': 'M',
  'SubAdd': 'Suite 6',
  'User_fld': '1'},
  'location': {'x': -73.73630468999994, 'y': 42.67997793600006},
  'score': 96.46},
 {'address': '80 S Swan St, Albany, NY, 12210',
  'attributes': {'Addr_type': 'StreetAddress',
  'Loc_name': '1A_SAM_AP_ZipN',
  'Match_addr': '80 S Swan St, Albany, NY, 12210',
  'ResultID': 1,
  'Score': 100,
  'Side': '',
  'Status': 'M',
  'SubAdd': '',
  'User_fld': '1'},
  'location': {'x': -73.75989873699996, 'y': 42.65406947400004},
  'score': 100}],
'spatialReference': {'latestWkid': 4326, 'wkid': 4326}}
```

3. For the three addresses contained in the batch you can see that more information is provided with the "Geocode Addresses" or batch functionality, specifically the individual locator is identified from the pool in the composite.

Batch Geocoding w/ ESRI API for Python (Jupyter Notebook)

The `arcgis.gis`, `arcpy` and other proprietary ESRI python libraries typically are not accessible outside of the ArcMAP python command line. However, ESRI has made its python libraries available through a python API that can be access by the Jupyter Notebook. Currently the Jupyter Notebook is the only method for accessing the API for Python outside of ArcMap session, and specific functionality is generated using `arcgis.gis` module and by establishing a connection with ArcGIS Online.

Note: I am still investigating this method and I have a question posted to the ESRI developer community because I do not feel the geocoding capabilities through the ArcGIS API for JavaScript are functioning as intended. The error/functionality question can be found in both the ESRI API for Python forum [here](#), and in the State/Local Government forum [here](#).

Sample Code:

```
In [1]: # Import the necessary libraries
```

```
    from arcgis.gis import GIS
```

```
    from arcgis.geocoding import Geocoder, get_geocoders, batch_geocode, geocode
```

```
In [2]: #Set the url for the Geocoder
```

```
    nys_gcdr_url =
```

```
    'https://gisservices.its.ny.gov/arcgis/rest/services/Locators/Street_and_Address_Composite/GeocodeServer'
```

```
In [3]: #Establish a connection to ArcGIS.com
```

```
    gis = GIS("http://www.arcgis.com", "username", "password ")
```

```
In [4]: #Set the geocoder to use to the NYS url established in "In [2]"
```

```
    esrinl_geocoder = Geocoder(nys_gcdr_url, gis)
```

```
In [5]: #Create your batch address list
```

```
    addresses = ["80 South Swan Street, Albany, NY, 12210",
```

```
                "10 B Airline Drive, Albany, NY, 12205",
```

```
                "800 North Pearl Street, Albany, NY, 12204"]
```

```
In [6]: #geocode the addresses
```

```
    results = batch_geocode(addresses, esrinl_geocoder)
```

In [7]: #Generate the match scores for each address against the geocoder

for result in results:

```
print("Score " + str(result['score']) + " : " + result['address'])
```

Out [7]: Score 99.52 : 80 S Swan St, Albany, New York, 12210

Score 100 : 10 Airline Dr, Albany, New York, 12205

Score 100 : 800 N Pearl St, Albany, New York, 12204

In [8]: #See the x,y results for an address in the batch

```
results[1]['location']
```

Out [8]: {'x': -73.81899493299994, 'y': 42.73686765800005}

In [9]: #Generate a quick map and plot the addresses

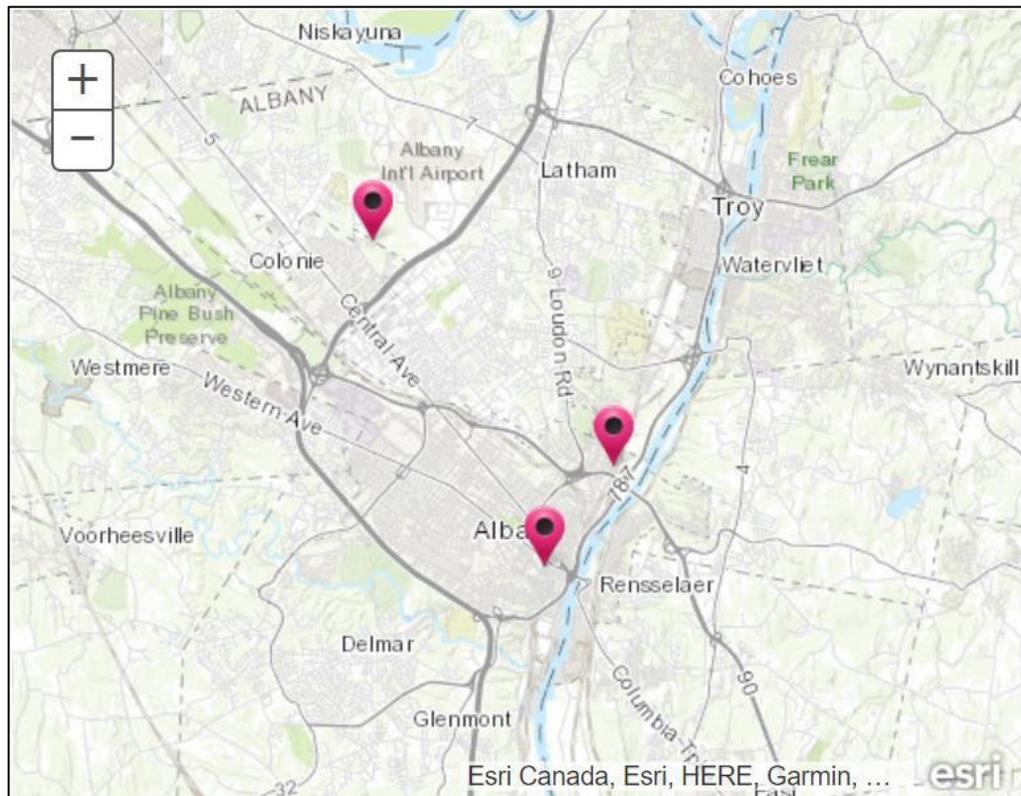
```
map = gis.map("New York", 6)
```

for address in results:

```
map.draw(address['location'])
```

```
map
```

Out [9]:





Geocoding with JavaScript/HTML

Coming Soon

The NYC Geographic Online Address Translator (GOAT)

The GOAT is maintained by the NYC Department of City Planning and was built specifically to work with Queens style addressing and was later expanded to work with the other four boroughs. It is important to note that the GOAT only geocodes to a street, and never to a point.

The GOAT can be accessed [here](#). The resulting GUI provides the user with an interface for entering address information for the NYC boroughs in various formats.

The screenshot shows the GOAT web application interface. At the top right is the NYC PLANNING logo. The main heading is "GOAT Geographic Online Address Translator". Below this is a welcome message and a description of the tool's capabilities. A navigation bar contains seven tabs: ADDRESS (Function 1B), INTERSECTION (Function 2), STREET SEGMENT (Function 3), STREET STRETCH (Function 3S), BLOCK & LOT (Function BL), BIN (Function BN), and STREET (Name/Code). The "ADDRESS" tab is selected. The main content area is titled "Display Street, Property Level and Address Point Information by Address" and includes a dropdown menu for "Select an Option Below" (currently set to "Display Street and Property Information"). Below this are input fields for "Select a Borough" (a dropdown menu), "Address Number", "Street or Place Name", and "Unit Input". A "Submit" button is located to the right of the "Street or Place Name" field. At the bottom, there are three columns of radio button options for "Normalize Input Street Name as:". The first column has "Roadbed Specific Information" (unchecked) and "TPAD" (checked). The second column has "Address Range List" (selected), "Complete BIN List" (unchecked), and "Display Both" (unchecked). The third column has "Input Street Name" (selected), "Primary Street Name" (unchecked), "Principal Street Name" (unchecked), and "DCP Preferred Street Name" (unchecked). A "Hide Search Options" link is located at the bottom right of the main content area.

GOAT Geographic Online Address Translator

Welcome to the Geographic Online Address Translator (GOAT)!

GOAT allows you to enter a New York City geographic location, such as an address, intersection, street segment, street stretch, block and lot or BIN, and returns back related geographic information, such as cross streets, side of street, tax block and lot (AKA Parcel -ID), five-digit ZIP code, census tract and block, police precinct, community district and city council district.

Information on the functions can be found in the [GOAT User Guide](#). Click on the output field label for its definition in the [Glossary](#).

ADDRESS (Function 1B) | **INTERSECTION (Function 2)** | **STREET SEGMENT (Function 3)** | **STREET STRETCH (Function 3S)** | **BLOCK & LOT (Function BL)** | **BIN (Function BN)** | **STREET (Name/Code)**

Display Street, Property Level and Address Point Information by Address ?

Select an Option Below
Display Street and Property Information ▼

Select a Borough | Address Number | Street or Place Name | Submit

-- Borough -- | | |

Unit Input | [Hide Search Options](#)

Roadbed Specific Information | Address Range List | Input Street Name
 TPAD | Complete BIN List | Primary Street Name
 Display Both | Principal Street Name | DCP Preferred Street Name

The GOAT provides several functions to the user:

- **Address (Function 1B)** – The input data for Function 1B is an address or a place name. Two sets of geographic information are returned; political geography associated with the entire block face containing the input location, followed by property level and building specific output.

The screenshot shows the 'ADDRESS (Function 1B)' tab selected. The main heading is 'Display Street, Property Level and Address Point Information by Address'. Below this, there is a dropdown menu set to 'Display Street and Property Information'. The search fields include 'Select a Borough' (set to '-- Borough --'), 'Address Number', 'Street or Place Name', and 'Unit Input'. A 'Submit' button is present. At the bottom, there are radio button options for 'Roadbed Specific Information' (unchecked), 'TPAD' (checked), 'Address Range List' (selected), 'Complete BIN List', 'Display Both', and 'Normalize Input Street Name as:' with options: 'Input Street Name' (selected), 'Primary Street Name', 'Principal Street Name', and 'DCP Preferred Street Name'. A 'Hide Search Options' link is also visible.

- **Intersection (Function 2)** – The input data to Function 2 is a street intersection. Function 2 returns geographic information related to the intersection, including the names of any additional streets that are at the intersection, and the administrative and political district identifiers within which the intersection is located. If an intersection lies on a boundary of two or more districts of a particular type, only one of those districts is listed. If the streets intersect twice, you must supply a Compass Direction from the dropdown list at the top of the screen. Please note that GOAT is unable to handle intersections if the streets intersect more than twice.

The screenshot shows the 'INTERSECTION (Function 2)' tab selected. The main heading is 'Display Geographic Information by Intersection'. The search fields include 'Select a Borough' (set to '-- Borough --'), 'Street 1 or Intersection Name', 'Street 2', and 'Compass Direction (if streets intersect twice)' (set to 'N/A'). A 'Submit' button is present. Below this, there is a section for 'Display Geographic Information By NODE ID' with a checkbox for 'Enter Node ID' and a text input field for 'Node ID'. A 'Hide Search Options' link is also visible. At the bottom, there are radio button options for 'Normalize Input Street Name as:' with options: 'Input Street Name' (selected), 'Primary Street Name', 'Principal Street Name', and 'DCP Preferred Street Name'.

- Street Segment (Function 3)** – Function 3 accepts an input street segment (the portion of a street between two consecutive cross streets) and returns related geographic information. If you are only interested in information for one side of the street segment (Block Face), you must supply the Side of Street from the dropdown list at the top of the screen. The information returned includes administrative and political district identifiers for the left and/or right side of the segment and the names of any additional cross streets that are present at the two endpoints of the segment.

- Street Stretch (Function 3S)** – Function 3S accepts as input a street stretch and returns the cross streets along the stretch. You may supply an ‘On’ street alone or an ‘On’ street and two cross streets, which will limit the amount of data returned. If you elect to supply cross streets and if either or both cross streets intersect the ‘On’ street exactly twice, you must supply a compass direction to determine which intersection you want selected. If one or both of the cross streets intersect the ‘On’ street more than twice, an error message will be returned indicating this problem and no data will be returned. If, on the other hand, you supply a compass direction and the ‘On’ street and cross street intersect once, in addition to the appropriate output, you will receive a warning message indicating that the compass direction was not required.

ADDRESS (Function 1B)	INTERSECTION (Function 2)	STREET SEGMENT (Function 3)	STREET STRETCH (Function 3S)	BLOCK & LOT (Function BL)	BIN (Function BN)	STREET (Name/Code)
--------------------------	------------------------------	--------------------------------	---------------------------------	------------------------------	----------------------	-----------------------

Display Street Stretch by "On" Street & Optional Two Cross Streets ?

Select a Borough:

On Street (Enter Street Name):

Show Real Streets Only

Compass Direction (optional):

First Cross Street (optional):

Compass Direction (optional):

Second Cross Street (optional):

- **Block & Lot (Function BL)** – The input data to Function BL is a Tax Block and Tax Lot. The information returned is building-specific and property-specific.

In addition to the 'fixed' data returned, you have a choice of receiving a list of up to twenty-one addresses which applies to the property or a list of BINs (Building Identification Numbers) that reside on the property.

ADDRESS (Function 1B)	INTERSECTION (Function 2)	STREET SEGMENT (Function 3)	STREET STRETCH (Function 3S)	BLOCK & LOT (Function BL)	BIN (Function BN)	STREET (Name/Code)
--------------------------	------------------------------	--------------------------------	---------------------------------	------------------------------	----------------------	-----------------------

Display Property Level Information by Block and Lot ?

Select a Borough:

Block:

Lot:

[Hide Search Options](#)

TPAD

Address Range List

Complete BIN List

Display Both

- **BIN (Function BIN)** – The input data to Function BN is a Building Identification Number (BIN). The information returned is building-specific and property-specific and represents information about the single building.

In addition to the 'fixed' data returned, you will receive a list of addresses which applies to the input BIN. The list of addresses should be complete because at this time no single building has more than twenty-one addresses.

ADDRESS (Function 1B)	INTERSECTION (Function 2)	STREET SEGMENT (Function 3)	STREET STRETCH (Function 3S)	BLOCK & LOT (Function BL)	BIN (Function BN)	STREET (Name/Code)
--------------------------	------------------------------	--------------------------------	---------------------------------	------------------------------	----------------------	-----------------------

Display Property Level Information by Building Identification Number (BIN) ?

Building Identification Number (BIN):

TPAD

- **Street (Function Name/Code)** – This function allows for a variety of options. A user can enter a street name, a street code, browse the Street Name Dictionary (SND), or normalize a street name. There are options to set the street name length, to choose Sort or Compact format (with the exception of Browse Street Name Dictionary (BB, BF) which always outputs street names in sort format), and to normalize the input street name to return the primary, principal or preferred name. Output, depending on the options chosen, includes Borough, Street Name, Street Code and Geographic Feature Type.

The screenshot shows the 'STREET (Name/Code)' function interface. At the top, there is a navigation bar with tabs for ADDRESS (Function 1B), INTERSECTION (Function 2), STREET SEGMENT (Function 3), STREET STRETCH (Function 3S), BLOCK & LOT (Function BL), BIN (Function BN), and STREET (Name/Code). The 'STREET (Name/Code)' tab is selected. Below the navigation bar, the main content area is titled 'Street Name/Code Information' with a help icon. It contains a form with the following elements: a dropdown menu labeled 'Select an Option Below' with the selected option 'Convert Street Name to Street Code (1N)'; a dropdown menu labeled 'Select a Borough' with the selected option '-- Borough --'; a text input field labeled 'Street Name'; a 'Submit' button; and a link labeled 'Hide Search Options'. Below the form, there are two sections: 'Street Name Length:' with a dropdown menu showing '32', and 'Street Name Format:' with two radio button options: 'Sort Format' (which is selected) and 'Compact Format'.

More detailed information on how to properly use each of these functions and on the GOAT in general can be found in the [Goat User Guide](#).



Grasshopper Open Source Geocoder

[Github Repository](#)