GeoPy Documentation

Release 1.23.0

GeoPy Contributors

Jul 04, 2020

Contents

Documentation https://geopy.readthedocs.io/

Source Code https://github.com/geopy/geopy

Issue Tracker https://github.com/geopy/geopy/issues

Stack Overflow https://stackoverflow.com/questions/tagged/geopy

PyPI https://pypi.org/project/geopy/

geopy is a Python 2 and 3 client for several popular geocoding web services.

geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

geopy is tested against CPython (versions 2.7, 3.4, 3.5, 3.6, 3.7, 3.8), PyPy, and PyPy3. geopy does not and will not support CPython 2.6.

CHAPTER 1

Installation

```
pip install geopy
```

CHAPTER 2

geopy 2.0

The 1.x series will not receive any new features or bugfixes unless explicitly asked on the issue tracker.

Consider upgrading to geopy 2.0.

CHAPTER 3

Geocoders

Each geolocation service you might use, such as Google Maps, Bing Maps, or Nominatim, has its own class in `geopy.geocoders` abstracting the service's API. Geocoders each define at least a `geocode` method, for resolving a location from a string, and may define a `reverse` method, which resolves a pair of coordinates to an address. Each Geocoder accepts any credentials or settings needed to interact with its service, e.g., an API key or locale, during its initialization.

To geolocate a query to an address and coordinates:

```
>>> from geopy.geocoders import Nominatim
>>> geolocator = Nominatim(user_agent="specify_your_app_name_here")
>>> location = geolocator.geocode("175 5th Avenue NYC")
>>> print(location.address)
Flatiron Building, 175, 5th Avenue, Flatiron, New York, NYC, New York, ...
>>> print((location.latitude, location.longitude))
(40.7410861, -73.9896297241625)
>>> print(location.raw)
{'place_id': '9167009604', 'type': 'attraction', ...}
```

To find the address corresponding to a set of coordinates:

```
>>> from geopy.geocoders import Nominatim
>>> geolocator = Nominatim(user_agent="specify_your_app_name_here")
>>> location = geolocator.reverse("52.509669, 13.376294")
>>> print(location.address)
Potsdamer Platz, Mitte, Berlin, 10117, Deutschland, European Union
>>> print((location.latitude, location.longitude))
(52.5094982, 13.3765983)
>>> print(location.raw)
{'place_id': '654513', 'osm_type': 'node', ...}
```

Locators' `geocode` and `reverse` methods require the argument `query`, and also accept at least the argument `exactly_one`, which is `True` by default. Geocoders may have additional attributes, e.g., Bing accepts `user_location`, the effect of which is to bias results near that location. `geocode` and `reverse` methods may return three types of values:

- When there are no results found, returns None.
- When the method's exactly_one argument is True and at least one result is found, returns a *geopy.location.Location* object, which can be iterated over as:

  ```
  (address<String>, (latitude<Float>, longitude<Float>))
  ```

  Or can be accessed as location.address, location.latitude, location.longitude, location.altitude, and location.raw. The last contains the full geocoder's response for this result.
- When exactly_one is False, and there is at least one result, returns a list of *geopy.location.Location* objects, as above:

  ```
  [location, [...]]
  ```

If a service is unavailable or otherwise returns a non-OK response, or doesn't receive a response in the allotted timeout, you will receive one of the *Exceptions* detailed below.

3.1 Specifying Parameters Once

Geocoding methods accept a lot of different parameters, and you would probably want to specify some of them just once and not care about them later.

This is easy to achieve with Python's functools.partial():

```
>>> from functools import partial
>>> from geopy.geocoders import Nominatim

>>> geolocator = Nominatim(user_agent="specify_your_app_name_here")

>>> geocode = partial(geolocator.geocode, language="es")
>>> print(geocode("london"))
Londres, Greater London, Inglaterra, SW1A 2DX, Gran Bretaña
>>> print(geocode("paris"))
París, Isla de Francia, Francia metropolitana, Francia
>>> print(geocode("paris", language="en"))
Paris, Ile-de-France, Metropolitan France, France

>>> reverse = partial(geolocator.reverse, language="es")
>>> print(reverse("52.509669, 13.376294"))
Steinecke, Potsdamer Platz, Tiergarten, Mitte, 10785, Alemania
```

If you need to modify the query, you can also use a one-liner with lambda. For example, if you only need to geocode locations in *Cleveland, Ohio*, you could do:

```
>>> geocode = lambda query: geolocator.geocode("%s, Cleveland OH" % query)
>>> print(geocode("11111 Euclid Ave"))
Thwing Center, Euclid Avenue, Magnolia-Wade Park Historic District,
University Circle, Cleveland, Cuyahoga County, Ohio, 44106, United States
of America
```

That lambda doesn't accept kwargs. If you need them, you could do:

```
>>> _geocode = partial(geolocator.geocode, language="es")
>>> geocode = lambda query, **kw: _geocode("%s, Cleveland OH" % query, **kw)
>>> print(geocode("11111 Euclid Ave"))
Thwing Center, Euclid Avenue, Magnolia-Wade Park Historic District,
University Circle, Cleveland, Cuyahoga County, Ohio, 44106, Estados Unidos
```

(continues on next page)

(continued from previous page)

```
>>> print(geocode("11111 Euclid Ave", language="en"))
Thwing Center, Euclid Avenue, Magnolia-Wade Park Historic District,
University Circle, Cleveland, Cuyahoga County, Ohio, 44106, United States
of America
```

3.2 Geopy Is Not a Service

Geocoding is provided by a number of different services, which are not affiliated with geopy in any way. These services provide APIs, which anyone could implement, and geopy is just a library which provides these implementations for many different services in a single package.

Therefore:

1. Different services have different Terms of Use, quotas, pricing, geodatabases and so on. For example, *geopy.geocoders.Nominatim* is free, but provides low request limits. If you need to make more queries, consider using another (probably paid) service, such as *geopy.geocoders.OpenMapQuest* or *geopy.geocoders.PickPoint* (these two are commercial providers of Nominatim, so they should have the same data and APIs). Or, if you are ready to wait, you can try *geopy.extra.rate_limiter.RateLimiter*.
2. geopy cannot be responsible for the geocoding services' databases. If you have issues with some queries which the service cannot fulfill, it should be directed to that service's support team.
3. geopy cannot be responsible for any networking issues between your computer and the geocoding service.

If you face any problem with your current geocoding service provider, you can always try a different one.

3.3 Accessing Geocoders

The typical way of retrieving a geocoder class is to make an import from `geopy.geocoders` package:

```
from geopy.geocoders import Nominatim
```

`geopy.geocoders.`**`get_geocoder_for_service`**`(`*service*`)`

For the service provided, try to return a geocoder class.

```
>>> from geopy.geocoders import get_geocoder_for_service
>>> get_geocoder_for_service("nominatim")
geopy.geocoders.osm.Nominatim
```

If the string given is not recognized, a *geopy.exc.GeocoderNotFound* exception is raised.

Given that almost all of the geocoders provide the `geocode` method it could be used to make basic queries based entirely on user input:

```
from geopy.geocoders import get_geocoder_for_service

def geocode(geocoder, config, query):
    cls = get_geocoder_for_service(geocoder)
    geolocator = cls(**config)
    location = geolocator.geocode(query)
    return location.address
```

(continues on next page)

(continued from previous page)

```
>>> geocode("nominatim", dict(user_agent="specify_your_app_name_here"), "london")
'London, Greater London, England, SW1A 2DX, United Kingdom'
>>> geocode("photon", dict(), "london")
'London, SW1A 2DX, London, England, United Kingdom'
```

3.4 Default Options Object

class geopy.geocoders.**options**

The *options* object contains default configuration values for geocoders, e.g. *timeout* and *User-Agent*. Instead of passing a custom value to each geocoder individually, you can override a default value in this object.

Please note that not all geocoders use all attributes of this object. For example, some geocoders don't respect the `default_scheme` attribute. Refer to the specific geocoder's initializer doc for a list of parameters which that geocoder accepts.

Example for overriding default `timeout` and `user_agent`:

```
>>> import geopy.geocoders
>>> from geopy.geocoders import Nominatim
>>> geopy.geocoders.options.default_user_agent = 'my_app/1'
>>> geopy.geocoders.options.default_timeout = 7
>>> geolocator = Nominatim()
>>> print(geolocator.headers)
{'User-Agent': 'my_app/1'}
>>> print(geolocator.timeout)
7
```

Attributes:

default_format_string String containing `'%s'` where the string to geocode should be interpolated before querying the geocoder. Used by *geocode* calls only. For example: `'%s, Mountain View, CA'`.

Deprecated since version 1.22.0: `format_string` is deprecated in favor of more advanced alternatives (see *Specifying Parameters Once*) and will be removed in geopy 2.0.

default_proxies Tunnel requests through HTTP proxy.

By default the system proxies are respected (e.g. *HTTP_PROXY* and *HTTPS_PROXY* env vars or platform-specific proxy settings, such as macOS or Windows native preferences – see `urllib.request.ProxyHandler` for more details). The *proxies* value for using system proxies is `None`.

To disable system proxies and issue requests directly, explicitly pass an empty dict as a value for *proxies*: `{}`.

To use a custom HTTP proxy location, pass a string. Valid examples are:

- `"192.0.2.0:8080"`
- `"john:passw0rd@192.0.2.0:8080"`
- `"http://john:passw0rd@192.0.2.0:8080"`

Please note:

- Scheme part (`http://`) of the proxy is ignored.

- Only *http* proxy is supported. Even if the proxy scheme is *https*, it will be ignored, and the connection between client and proxy would still be unencrypted. However, *https* requests via *http* proxy are still supported (via *HTTP CONNECT* method).

Raw urllib-style *proxies* dict might be provided instead of a string:

- `{"https": "192.0.2.0:8080"}` – means that HTTP proxy would be used only for requests having *https* scheme. String *proxies* value is automatically used for both schemes, and is provided as a shorthand for the urllib-style *proxies* dict.

For more information, see documentation on `urllib.request.ProxyHandler`.

Changed in version 1.15.0: Added support for the string value.

default_scheme Use `'https'` or `'http'` as the API URL's scheme.

default_ssl_context An `ssl.SSLContext` instance with custom TLS verification settings. Pass `None` to use the interpreter's defaults (starting from Python 2.7.9 and 3.4.3 that is to use the system's trusted CA certificates; the older versions don't support TLS verification completely).

For older versions of Python (before 2.7.9 and 3.4.3) this argument is ignored, as *urlopen* doesn't accept an ssl context there, and a warning is issued.

To use the CA bundle used by *requests* library:

```
import ssl
import certifi
import geopy.geocoders
ctx = ssl.create_default_context(cafile=certifi.where())
geopy.geocoders.options.default_ssl_context = ctx
```

To disable TLS certificate verification completely:

```
import ssl
import geopy.geocoders
ctx = ssl.create_default_context()
ctx.check_hostname = False
ctx.verify_mode = ssl.CERT_NONE
geopy.geocoders.options.default_ssl_context = ctx
```

See docs for the `ssl.SSLContext` class for more examples.

default_timeout Time, in seconds, to wait for the geocoding service to respond before raising a *`geopy.exc.GeocoderTimedOut`* exception. Pass *None* to disable timeout.

Note: Currently `None` as a value is processed correctly only for the `geopy.geocoders.options.default_timeout` option value. `timeout=None` as a method argument (i.e. `geocoder.geocode(..., timeout=None)`) would be treated as "use timeout, as set in `geopy.geocoders.options.default_timeout`", and a deprecation warning would be raised. In geopy 2.0 this will change, so that `timeout=None` would actually disable timeout.

default_user_agent User-Agent header to send with the requests to geocoder API.

default_format_string = '%s'

default_proxies = None

default_scheme = 'https'

default_ssl_context = None

default_timeout = 1

default_user_agent = 'geopy/1.23.0'

3.5 Usage with Pandas

It’s possible to geocode a pandas DataFrame with geopy, however, rate-limiting must be taken into account.

A large number of DataFrame rows might produce a significant amount of geocoding requests to a Geocoding service, which might be throttled by the service (e.g. by returning *Too Many Requests* 429 HTTP error or timing out).

`geopy.extra.rate_limiter.RateLimiter` class provides a convenient wrapper, which can be used to automatically add delays between geocoding calls to reduce the load on the Geocoding service. Also it can retry failed requests and swallow errors for individual rows.

If you’re having the *Too Many Requests* error, you may try the following:

- Use `geopy.extra.rate_limiter.RateLimiter` with non-zero `min_delay_seconds`.
- Try a different Geocoding service (please consult with their ToS first, as some services prohibit bulk geocoding).
- Take a paid plan on the chosen Geocoding service, which provides higher quota.
- Provision your own local copy of the Geocoding service (such as Nominatim).

class geopy.extra.rate_limiter.**RateLimiter**(*func*, *min_delay_seconds=0.0*, *max_retries=2*, *error_wait_seconds=5.0*, *swallow_exceptions=True*, *return_value_on_exception=None*)

RateLimiter allows to perform bulk operations while gracefully handling error responses and adding delays when needed.

In the example below a delay of 1 second (`min_delay_seconds=1`) will be added between each pair of `geolocator.geocode` calls; all `geopy.exc.GeocoderServiceError` exceptions will be retried (up to `max_retries` times):

```
import pandas as pd
df = pd.DataFrame({'name': ['paris', 'berlin', 'london']})

from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="specify_your_app_name_here")

from geopy.extra.rate_limiter import RateLimiter
geocode = RateLimiter(geolocator.geocode, min_delay_seconds=1)
df['location'] = df['name'].apply(geocode)

df['point'] = df['location'].apply(lambda loc: tuple(loc.point) if loc else None)
```

This would produce the following DataFrame:

```
>>> df
     name                                           location  \
0   paris  (Paris, Île-de-France, France métropolitaine, ...
1  berlin  (Berlin, 10117, Deutschland, (52.5170365, 13.3...
2  london  (London, Greater London, England, SW1A 2DU, UK...

                           point
0   (48.8566101, 2.3514992, 0.0)
```

(continues on next page)

(continued from previous page)

```
1   (52.5170365, 13.3888599, 0.0)
2   (51.5073219, -0.1276474, 0.0)
```

To pass extra options to the *geocode* call:

```
from functools import partial
df['location'] = df['name'].apply(partial(geocode, language='de'))
```

To see a progress bar:

```
from tqdm import tqdm
tqdm.pandas()
df['location'] = df['name'].progress_apply(geocode)
```

Before using this class, please consult with the Geocoding service ToS, which might explicitly consider bulk requests (even throttled) a violation.

New in version 1.16.0.

`__init__`(*func*, *min_delay_seconds=0.0*, *max_retries=2*, *error_wait_seconds=5.0*, *swallow_exceptions=True*, *return_value_on_exception=None*)

Parameters

- **`func`** (`callable`) – A function which should be wrapped by the `RateLimiter`.
- **`min_delay_seconds`** (`float`) – Minimum delay in seconds between the wrapped `func` calls.
- **`max_retries`** (`int`) – Number of retries on exceptions. Only `geopy.exc.GeocoderServiceError` exceptions are retried – others are always re-raised. `max_retries + 1` requests would be performed at max per query. Set `max_retries=0` to disable retries.
- **`error_wait_seconds`** (`float`) – Time to wait between retries after errors. Must be greater or equal to `min_delay_seconds`.
- **`swallow_exceptions`** (`bool`) – Should an exception be swallowed after retries? If not, it will be re-raised. If yes, the `return_value_on_exception` will be returned.
- **`return_value_on_exception`** – Value to return on failure when `swallow_exceptions=True`.

3.6 AlgoliaPlaces

class `geopy.geocoders.`**`AlgoliaPlaces`**(*app_id=None*, *api_key=None*, *domain='places-dsn.algolia.net'*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the Algolia Places API.

Documentation at: https://community.algolia.com/places/documentation.html

New in version 1.22.0.

`__init__`(*app_id=None*, *api_key=None*, *domain='places-dsn.algolia.net'*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **app_id** (*str*) – Unique application identifier. It's used to identify you when using Algolia's API. See https://www.algolia.com/dashboard.
- **api_key** (*str*) – Algolia's user API key.
- **domain** (*str*) – Currently it is `'places-dsn.algolia.net'`, can be changed for testing purposes.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *type=None*, *restrict_searchable_attributes=None*, *limit=None*, *language=None*, *countries=None*, *around=None*, *around_via_ip=None*, *around_radius=None*, *x_forwarded_for=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **type** (*str*) – Restrict the search results to a specific type. Available types are defined in documentation: https://community.algolia.com/places/api-clients.html#api-options-type
- **restrict_searchable_attributes** (*str*) – Restrict the fields in which the search is done.
- **limit** (*int*) – Limit the maximum number of items in the response. If not provided and there are multiple results Algolia API will return 20 results by default. This will be reset to one if `exactly_one` is True.
- **language** (*str*) – If specified, restrict the search results to a single language. You can pass two letters country codes (ISO 639-1).
- **countries** (*list*) – If specified, restrict the search results to a specific list of countries. You can pass two letters country codes (ISO 3166-1).
- **around** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – Force to first search around a specific latitude longitude.
- **around_via_ip** (*bool*) – Whether or not to first search around the geolocation of the user found via his IP address. This is true by default.

- **around_radius** – Radius in meters to search around the latitude/longitude. Otherwise a default radius is automatically computed given the area density.
- **x_forwarded_for** (*str*) – Override the HTTP header X-Forwarded-For. With this you can control the source IP address used to resolve the geo-location of the user. This is particularly useful when you want to use the API from your backend as if it was from your end-users' locations.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *language=None*)
Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s".) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Limit the maximum number of items in the response. If not provided and there are multiple results Algolia API will return 20 results by default. This will be reset to one if exactly_one is True.
- **language** (*str*) – If specified, restrict the search results to a single language. You can pass two letters country codes (ISO 639-1).

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

3.7 ArcGIS

class geopy.geocoders.**ArcGIS**(*username=None*, *password=None*, *referer=None*, *token_lifetime=60*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *auth_domain='www.arcgis.com'*, *domain='geocode.arcgis.com'*)
Geocoder using the ERSI ArcGIS API.

Documentation at: https://developers.arcgis.com/rest/geocode/api-reference/overview-world-geocoding-service.htm

__init__(*username=None*, *password=None*, *referer=None*, *token_lifetime=60*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *auth_domain='www.arcgis.com'*, *domain='geocode.arcgis.com'*)

Parameters

- **username** (*str*) – ArcGIS username. Required if authenticated mode is desired.
- **password** (*str*) – ArcGIS password. Required if authenticated mode is desired.

- **referer** (*str*) – Required if authenticated mode is desired. *Referer* HTTP header to send with each request, e.g., `'http://www.example.com'`. This is tied to an issued token, so fielding queries for multiple referrers should be handled by having multiple ArcGIS geocoder instances.
- **token_lifetime** (*int*) – Desired lifetime, in minutes, of an ArcGIS-issued token.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*. If authenticated mode is in use, it must be `'https'`.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.
- **auth_domain** (*str*) – Domain where the target ArcGIS auth service is hosted. Used only in authenticated mode (i.e. username, password and referer are set).

  New in version 1.17.0.
- **domain** (*str*) – Domain where the target ArcGIS service is hosted.

  New in version 1.17.0.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *out_fields=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **out_fields** (*str or iterable*) – A list of output fields to be returned in the attributes field of the raw data. This can be either a python list/tuple of fields or a comma-separated string. See https://developers.arcgis.com/rest/geocode/api-reference/geocoding-service-output.htm for a list of supported output fields. If you want to return all supported output fields, set `out_fields="*"`.

  New in version 1.14.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *distance=None*, *wkid=4326*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **distance** (*int*) – Distance from the query location, in meters, within which to search. ArcGIS has a default of 100 meters, if not specified.
- **wkid** (*str*) – WKID to use for both input and output coordinates.

  Deprecated since version 1.14.0: It wasn't working before because it was specified incorrectly in the request parameters, and won't work even if we fix the request, because *geopy.point.Point* normalizes the coordinates according to WKID 4326. Please open an issue in the geopy issue tracker if you believe that custom wkid values should be supported. This parameter is scheduled for removal in geopy 2.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.8 AzureMaps

class `geopy.geocoders.`**AzureMaps**(*subscription_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='atlas.microsoft.com'*)

Bases: `geopy.geocoders.tomtom.TomTom`

AzureMaps geocoder based on TomTom.

Documentation at: https://docs.microsoft.com/en-us/azure/azure-maps/index

New in version 1.15.0.

__init__(*subscription_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='atlas.microsoft.com'*)

Parameters

- **subscription_key** (*str*) – Azure Maps subscription key.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.
- **domain** (*str*) – Domain where the target Azure Maps service is hosted.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *typeahead=False*, *language=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Maximum amount of results to return from the service. Unless exactly_one is set to False, limit will always be 1.
- **typeahead** (*bool*) – If the "typeahead" flag is set, the query will be interpreted as a partial input and the search will enter predictive mode.
- **language** (*str*) – Language in which search results should be returned. When data in specified language is not available for a specific field, default language is used. List of supported languages (case-insensitive): https://developer.tomtom.com/online-search/online-search-documentation/supported-languages

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=None*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s".) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Language in which search results should be returned. When data in specified language is not available for a specific field, default language is used. List of supported languages (case-insensitive): https://developer.tomtom.com/online-search/online-search-documentation/supported-languages

  New in version 1.18.0.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

3.9 Baidu

class geopy.geocoders.**Baidu**(*api_key*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *security_key=None*)

Geocoder using the Baidu Maps v2 API.

Documentation at: http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding

Attention: Newly registered API keys will not work with v2 API, use *BaiduV3* instead.

New in version 1.0.0.

__init__(*api_key*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *security_key=None*)

Parameters

- **api_key** (*str*) – The API key (AK) required by Baidu Map to perform geocoding requests. API keys are managed through the Baidu APIs console (http://lbsyun.baidu.com/apiconsole/key).
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.

  Changed in version 1.14.0: Default scheme has been changed from `http` to `https`.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.
- **security_key** (*str*) – The security key (SK) to calculate the SN parameter in request if authentication setting requires (http://lbsyun.baidu.com/index.php?title=lbscloud/api/appendix).

  New in version 1.15.0.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available. Baidu's API will always return at most one result.

  New in version 1.14.0.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.10 BaiduV3

class `geopy.geocoders.`**BaiduV3**(*api_key*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *security_key=None*)

Bases: `geopy.geocoders.baidu.Baidu`

Geocoder using the Baidu Maps v3 API.

Documentation at: http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding

New in version 1.22.0.

__init__(*api_key*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*, *security_key=None*)

Parameters

- **api_key** (*str*) – The API key (AK) required by Baidu Map to perform geocoding requests. API keys are managed through the Baidu APIs console (http://lbsyun.baidu.com/apiconsole/key).
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.

  Changed in version 1.14.0: Default scheme has been changed from `http` to `https`.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

New in version 1.12.0.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.

- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

- **security_key** (*str*) – The security key (SK) to calculate the SN parameter in request if authentication setting requires (http://lbsyun.baidu.com/index.php?title=lbscloud/api/appendix).

  New in version 1.15.0.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available. Baidu's API will always return at most one result.

  New in version 1.14.0.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.11 BANFrance

class geopy.geocoders.**BANFrance**(*domain='api-adresse.data.gouv.fr', format_string=None, scheme=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, ssl_context=DEFAULT_SENTINEL*)

Geocoder using the Base Adresse Nationale France API.

Documentation at: https://adresse.data.gouv.fr/api

New in version 1.18.0.

__init__(*domain='api-adresse.data.gouv.fr', format_string=None, scheme=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, ssl_context=DEFAULT_SENTINEL*)

Parameters

- **domain** (*str*) – Currently it is `'api-adresse.data.gouv.fr'`, can be changed for testing purposes.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

geocode(*query, limit=None, exactly_one=True, timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **limit** (*int*) – Defines the maximum number of items in the response structure. If not provided and there are multiple results the BAN API will return 5 results by default. This will be reset to one if `exactly_one` is True.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query, exactly_one=True, timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.12 Bing

class geopy.geocoders.**Bing**(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the Bing Maps Locations API.

Documentation at: https://msdn.microsoft.com/en-us/library/ff701715.aspx

__init__(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – Should be a valid Bing Maps API key (https://www.microsoft.com/en-us/maps/create-a-bing-maps-key).
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode(*query*, *exactly_one=True*, *user_location=None*, *timeout=DEFAULT_SENTINEL*, *culture=None*, *include_neighborhood=None*, *include_country_code=False*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.

  For a structured query, provide a dictionary whose keys are one of: *addressLine*, *locality* (city), *adminDistrict* (state), *countryRegion*, or *postalcode*.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **user_location** (*geopy.point.Point*) – Prioritize results closer to this location.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **culture** (*str*) – Affects the language of the response, must be a two-letter country code.

  New in version 1.4.0.
- **include_neighborhood** (*bool*) – Sets whether to include the neighborhood field in the response.

  New in version 1.4.0.
- **include_country_code** (*bool*) – Sets whether to include the two-letter ISO code of the country in the response (field name 'countryRegionIso2').

  New in version 1.4.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *culture=None*, *include_country_code=False*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **culture** (*str*) – Affects the language of the response, must be a two-letter country code.
- **include_country_code** (*bool*) – Sets whether to include the two-letter ISO code of the country in the response (field name 'countryRegionIso2').

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.13 DataBC

class geopy.geocoders.**DataBC** (*scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the Physical Address Geocoder from DataBC.

Documentation at: http://www.data.gov.bc.ca/dbc/geographic/locate/geocoding.page

__init__ (*scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode(*query*, *max_results=25*, *set_back=0*, *location_descriptor='any'*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **max_results** (*int*) – The maximum number of resutls to request.
- **set_back** (*float*) – The distance to move the accessPoint away from the curb (in meters) and towards the interior of the parcel. location_descriptor must be set to accessPoint for set_back to take effect.
- **location_descriptor** (*str*) – The type of point requested. It can be any, accessPoint, frontDoorPoint, parcelPoint, rooftopPoint and routingPoint.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

3.14 GeocodeEarth

class geopy.geocoders.**GeocodeEarth**(*api_key*, *format_string=None*, *boundary_rect=None*, *country_bias=None*, *domain='api.geocode.earth'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *scheme=None*, *ssl_context=DEFAULT_SENTINEL*)

Bases: geopy.geocoders.pelias.Pelias

geocode.earth, a Pelias-based service provided by the developers of Pelias itself.

New in version 1.15.0.

`__init__`(*api_key*, *format_string=None*, *boundary_rect=None*, *country_bias=None*, *domain='api.geocode.earth'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *scheme=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – Geocode.earth API key, required.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **boundary_rect** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.17.0: Previously boundary_rect could be a list of 4 strings or numbers in the format of `[longitude, latitude, longitude, latitude]`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *boundary_rect* instead.
- **country_bias** (*str*) – Bias results to this country (ISO alpha-3).

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_bias* instead.
- **domain** (*str*) – Specify a custom domain for Pelias API.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *boundary_rect=None*, *country_bias=None*, *language=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address, query or structured query to geocode you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **boundary_rect** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.19.0.

- **country_bias** (*str*) – Bias results to this country (ISO alpha-3).

  New in version 1.19.0.

- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.21.0.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=None*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s".) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.21.0.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

3.15 GeocodeFarm

class geopy.geocoders.**GeocodeFarm**(*api_key=None*, *format_string=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *scheme=None*)

Geocoder using the GeocodeFarm API.

Documentation at: https://www.geocode.farm/geocoding/free-api-documentation/

__init__(*api_key=None*, *format_string=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *scheme=None*)

Parameters

- **api_key** (*str*) – (optional) The API key required by GeocodeFarm to perform geocoding requests.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.

- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.

- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.

  New in version 1.14.0.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available. GeocodeFarm's API will always return at most one result.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.16 Geolake

class geopy.geocoders.**Geolake**(*api_key*, *domain='api.geolake.com'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the Geolake API.

Documentation at: https://geolake.com/docs/api

Terms of Service at: https://geolake.com/terms-of-use

New in version 1.18.0.

`__init__`(*api_key*, *domain='api.geolake.com'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **`format_string`** (*`str`*) – See *`geopy.geocoders.options.default_format_string`*.

  Deprecated since version 1.22.0.

- **`api_key`** (*`str`*) – The API key required by Geolake to perform geocoding requests. You can get your key here: https://geolake.com/
- **`domain`** (*`str`*) – Currently it is `'api.geolake.com'`, can be changed for testing purposes.
- **`scheme`** (*`str`*) – See *`geopy.geocoders.options.default_scheme`*.
- **`timeout`** (*`int`*) – See *`geopy.geocoders.options.default_timeout`*.
- **`proxies`** (*`dict`*) – See *`geopy.geocoders.options.default_proxies`*.
- **`user_agent`** (*`str`*) – See *`geopy.geocoders.options.default_user_agent`*.
- **`ssl_context`** (`ssl.SSLContext`) – See *`geopy.geocoders.options.default_ssl_context`*.

`geocode`(*query*, *country_codes=None*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **`query`** (*`str`*) – The address or query you wish to geocode.

  For a structured query, provide a dictionary whose keys are one of: *country*, *state*, *city*, *zipcode*, *street*, *address*, *houseNumber* or *subNumber*.

- **`country_codes`** (*`str or list`*) – Provides the geocoder with a list of country codes that the query may reside in. This value will limit the geocoder to the supplied countries. The country code is a 2 character code as defined by the ISO-3166-1 alpha-2 standard (e.g. `FR`). Multiple countries can be specified with a Python list.

  Changed in version 1.19.0: Previously only a Python list of countries could be specified. Now a single country as a string can be specified as well.

- **`exactly_one`** (*`bool`*) – Return one result or a list of one result.
- **`timeout`** (*`int`*) – Time, in seconds, to wait for the geocoding service to respond before raising a *`geopy.exc.GeocoderTimedOut`* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *`geopy.location.Location`* or a list of them, if `exactly_one=False`.

3.17 GeoNames

class geopy.geocoders.**GeoNames**(*country_bias=None, username=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, format_string=None, ssl_context=DEFAULT_SENTINEL, scheme='http'*)

GeoNames geocoder.

Documentation at: http://www.geonames.org/export/geonames-search.html

Reverse geocoding documentation at: http://www.geonames.org/export/web-services.html#findNearbyPlaceName

__init__(*country_bias=None, username=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, format_string=None, ssl_context=DEFAULT_SENTINEL, scheme='http'*)

Parameters

- **country_bias** (*str*) – Records from the country_bias are listed first. Two letter country code ISO-3166.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_bias* instead.

- **username** (*str*) – GeoNames username, required. Sign up here: http://www.geonames.org/login

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.

- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.

- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.

- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*. Note that at the time of writing GeoNames doesn't support *https*, so the default scheme is *http*. The value of *geopy.geocoders.options.default_scheme* is not respected. This parameter is present to make it possible to switch to *https* once GeoNames adds support for it.

  New in version 1.18.0.

geocode(*query, exactly_one=True, timeout=DEFAULT_SENTINEL, country=None, country_bias=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.

- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **country** (*str or list*) – Limit records to the specified countries. Two letter country code ISO-3166 (e.g. `FR`). Might be a single string or a list of strings.

  New in version 1.19.0.
- **country_bias** (*str*) – Records from the country_bias are listed first. Two letter country code ISO-3166.

  New in version 1.19.0.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=DEFAULT_SENTINEL*, *timeout=DEFAULT_SENTINEL*, *feature_code=None*, *lang=None*, *find_nearby_type='findNearbyPlaceName'*)
Return an address by location point.

New in version 1.2.0.

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.14.0: Default value for `exactly_one` was `False`, which differs from the conventional default across geopy. Please always pass this argument explicitly, otherwise you would get a warning. In geopy 2.0 the default value will become `True`.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **feature_code** (*str*) – A GeoNames feature code

  New in version 1.18.0.
- **lang** (*str*) – language of the returned `name` element (the pseudo language code 'local' will return it in local language) Full list of supported languages can be found here: https://www.geonames.org/countries/

  New in version 1.18.0.
- **find_nearby_type** (*str*) – A flag to switch between different GeoNames API endpoints. The default value is `findNearbyPlaceName` which returns the closest populated place. Another currently implemented option is `findNearby` which returns the closest toponym for the lat/lng query.

  New in version 1.18.0.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse_timezone(*query*, *timeout=DEFAULT_SENTINEL*)
Find the timezone for a point in *query*.

GeoNames always returns a timezone: if the point being queried doesn't have an assigned Olson timezone id, a `pytz.FixedOffset` timezone is used to produce the `geopy.timezone.Timezone`.

New in version 1.18.0.

Parameters

- **`query`** (`geopy.point.Point`, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s") – The coordinates for which you want a timezone.
- **`timeout`** (`int`) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `geopy.timezone.Timezone`

3.18 GoogleV3

class `geopy.geocoders.`**`GoogleV3`**(*api_key=None, domain='maps.googleapis.com', scheme=None, client_id=None, secret_key=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, format_string=None, ssl_context=DEFAULT_SENTINEL, channel=''*)

Geocoder using the Google Maps v3 API.

Documentation at: https://developers.google.com/maps/documentation/geocoding/

Attention: Since July 2018 Google requires each request to have an API key. See https://developers.google.com/maps/documentation/geocoding/usage-and-billing

`__init__`(*api_key=None, domain='maps.googleapis.com', scheme=None, client_id=None, secret_key=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, format_string=None, ssl_context=DEFAULT_SENTINEL, channel=''*)

Parameters

- **`api_key`** (`str`) – The API key required by Google to perform geocoding requests. API keys are managed through the Google APIs console (https://code.google.com/apis/console). Make sure to have both `Geocoding API` and `Time Zone API` services enabled for this API key.
- **`domain`** (`str`) – Should be the localized Google Maps domain to connect to. The default is 'maps.googleapis.com', but if you're geocoding address in the UK (for example), you may want to set it to 'maps.google.co.uk' to properly bias results.
- **`scheme`** (`str`) – See `geopy.geocoders.options.default_scheme`.
- **`client_id`** (`str`) – If using premier, the account client id.
- **`secret_key`** (`str`) – If using premier, the account secret key.
- **`timeout`** (`int`) – See `geopy.geocoders.options.default_timeout`.
- **`proxies`** (`dict`) – See `geopy.geocoders.options.default_proxies`.
- **`user_agent`** (`str`) – See `geopy.geocoders.options.default_user_agent`.

  New in version 1.12.0.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.

- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

- **channel** (*str*) – If using premier, the channel identifier.

  New in version 1.12.0.

geocode(*query=None*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *bounds=None*, *region=None*, *components=None*, *place_id=None*, *language=None*, *sensor=False*)
Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode. Optional, if `components` param is set:

```
>>> g.geocode(components={"city": "Paris", "country": "FR"})
Location(France, (46.227638, 2.213749, 0.0))
```

  Changed in version 1.14.0: Now `query` is optional if `components` param is set.

- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

- **bounds** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – The bounding box of the viewport within which to bias geocode results more prominently. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.17.0: Previously the only supported format for bounds was a list like `[latitude, longitude, latitude, longitude]`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.

- **region** (*str*) – The region code, specified as a ccTLD ("top-level domain") two-character value.

- **components** (*dict or list*) – Restricts to an area. Can use any combination of: *route*, *locality*, *administrative_area*, *postal_code*, *country*.

  Pass a list of tuples if you want to specify multiple components of the same type, e.g.:

```
>>> [('administrative_area', 'VA'), ('administrative_area',
↪'Arlington')]
```

  Changed in version 1.22.0: Added support for a list of tuples.

- **place_id** (*str*) – Retrieve a Location using a Place ID. Cannot be not used with `query` or `bounds` parameters.

```
>>> g.geocode(place_id='ChIJOcfP0Iq2j4ARDrXUa7ZWs34')
```

New in version 1.19.0.

- **language** (*str*) – The language in which to return results.
- **sensor** (*bool*) – Whether the geocoding request comes from a device with a location sensor.

Return type None, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=DEFAULT_SENTINEL*, *timeout=DEFAULT_SENTINEL*, *language=None*, *sensor=False*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.14.0: Default value for `exactly_one` was `False`, which differs from the conventional default across geopy. Please always pass this argument explicitly, otherwise you would get a warning. In geopy 2.0 the default value will become `True`.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – The language in which to return results.
- **sensor** (*bool*) – Whether the geocoding request comes from a device with a location sensor.

Return type None, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse_timezone(*query*, *at_time=None*, *timeout=DEFAULT_SENTINEL*)

Find the timezone a point in *query* was in for a specified *at_time*.

New in version 1.18.0.

Changed in version 1.18.1: Previously a `KeyError` was raised for a point without an assigned Olson timezone id (e.g. for Antarctica). Now this method returns None for such requests.

Parameters

- **query** (*geopy.point.Point*, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s") – The coordinates for which you want a timezone.
- **at_time** (`datetime.datetime` or None) – The time at which you want the timezone of this location. This is optional, and defaults to the time that the function is called in UTC. Timezone-aware datetimes are correctly handled and naive datetimes are silently treated as UTC.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type None or *geopy.timezone.Timezone*

timezone(*location*, *at_time=None*, *timeout=DEFAULT_SENTINEL*)

Find the timezone a *location* was in for a specified *at_time*, and return a pytz timezone object.

New in version 1.2.0.

Deprecated since version 1.18.0: Use `GoogleV3.reverse_timezone()` instead. This method will be removed in geopy 2.0.

Changed in version 1.18.1: Previously a `KeyError` was raised for a point without an assigned Olson timezone id (e.g. for Antarctica). Now this method returns None for such requests.

Parameters

- **location** (*geopy.point.Point*, list or tuple of (latitude, longitude), or string as "%(latitude)s, %(longitude)s") – The coordinates for which you want a timezone.
- **at_time** (`datetime.datetime` or None) – The time at which you want the timezone of this location. This is optional, and defaults to the time that the function is called in UTC. Timezone-aware datetimes are correctly handled and naive datetimes are silently treated as UTC.

  Changed in version 1.18.0: Previously this parameter accepted raw unix timestamp as int or float. This is now deprecated in favor of datetimes and support for numbers will be removed in geopy 2.0.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None` or pytz timezone. See `pytz.timezone()`.

3.19 HERE

class geopy.geocoders.**Here**(*app_id=None, app_code=None, apikey=None, format_string=None, scheme=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, ssl_context=DEFAULT_SENTINEL*)

Geocoder using the HERE Geocoder API.

Documentation at: https://developer.here.com/documentation/geocoder/

New in version 1.15.0.

__init__(*app_id=None, app_code=None, apikey=None, format_string=None, scheme=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, ssl_context=DEFAULT_SENTINEL*)

Parameters

- **app_id** (*str*) – Should be a valid HERE Maps APP ID. Will eventually be replaced with APIKEY. See https://developer.here.com/authenticationpage.

  Deprecated since version 1.21.0: App ID and App Code are being replaced by API Keys and OAuth 2.0 by HERE. Consider getting an `apikey` instead of using `app_id` and `app_code`.
- **app_code** (*str*) – Should be a valid HERE Maps APP CODE. Will eventually be replaced with APIKEY. See https://developer.here.com/authenticationpage.

  Deprecated since version 1.21.0.
- **apikey** (*str*) – Should be a valid HERE Maps APIKEY. These keys were introduced in December 2019 and will eventually replace the legacy APP CODE/APP ID pairs which are already no longer available for new accounts (but still work

for old accounts). More authentication details are available at https://developer.here.com/blog/announcing-two-new-authentication-types. See https://developer.here.com/authenticationpage.

New in version 1.21.0.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.

- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

geocode(*query*, *bbox=None*, *mapview=None*, *exactly_one=True*, *maxresults=None*, *pageinformation=None*, *language=None*, *additional_data=False*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

This implementation supports only a subset of all available parameters. A list of all parameters of the pure REST API is available here: https://developer.here.com/documentation/geocoder/topics/resource-geocode.html

Parameters

- **query** (*str*) – The address or query you wish to geocode.

  For a structured query, provide a dictionary whose keys are one of: *city*, *county*, *district*, *country*, *state*, *street*, *housenumber*, or *postalcode*.

- **bbox** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – A type of spatial filter, limits the search for any other attributes in the request. Specified by two coordinate (lat/lon) pairs – corners of the box. *The bbox search is currently similar to mapview but it is not extended* (cited from the REST API docs). Relevant global results are also returned. Example: `[Point(22, 180), Point(-22, -180)]`.
- **mapview** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – The app's viewport, given as two coordinate pairs, specified by two lat/lon pairs – corners of the bounding box, respectively. Matches from within the set map view plus an extended area are ranked highest. Relevant global results are also returned. Example: `[Point(22, 180), Point(-22, -180)]`.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **maxresults** (*int*) – Defines the maximum number of items in the response structure. If not provided and there are multiple results the HERE API will return 10 results by default. This will be reset to one if `exactly_one` is True.
- **pageinformation** (*int*) – A key which identifies the page to be returned when the response is separated into multiple pages. Only useful when `maxresults` is also provided.
- **language** (*str*) – Affects the language of the response, must be a RFC 4647 language code, e.g. 'en-US'.

- **additional_data** (*str*) – A string with key-value pairs as described on https://developer.here.com/documentation/geocoder/topics/resource-params-additional.html. These will be added as one query parameter to the URL.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse(*query*, *radius=None*, *exactly_one=True*, *maxresults=None*, *pageinformation=None*, *language=None*, *mode='retrieveAddresses'*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

This implementation supports only a subset of all available parameters. A list of all parameters of the pure REST API is available here: https://developer.here.com/documentation/geocoder/topics/resource-reverse-geocode.html

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **radius** (*float*) – Proximity radius in meters.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **maxresults** (*int*) – Defines the maximum number of items in the response structure. If not provided and there are multiple results the HERE API will return 10 results by default. This will be reset to one if `exactly_one` is True.
- **pageinformation** (*int*) – A key which identifies the page to be returned when the response is separated into multiple pages. Only useful when `maxresults` is also provided.
- **language** (*str*) – Affects the language of the response, must be a RFC 4647 language code, e.g. 'en-US'.
- **mode** (*str*) – Affects the type of returned response items, must be one of: 'retrieveAddresses' (default), 'retrieveAreas', 'retrieveLandmarks', 'retrieveAll', or 'trackPosition'. See online documentation for more information.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.20 IGNFrance

class geopy.geocoders.**IGNFrance**(*api_key*, *username=None*, *password=None*, *referer=None*, *domain='wxs.ign.fr'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the IGN France GeoCoder OpenLS API.

Documentation at: https://geoservices.ign.fr/documentation/geoservices/index.html

`__init__`(*api_key*, *username=None*, *password=None*, *referer=None*, *domain='wxs.ign.fr'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **`api_key`** (*`str`*) – The API key required by IGN France API to perform geocoding requests. You can get your key here: https://geoservices.ign.fr/documentation/services-acces.html. Mandatory. For authentication with referer and with username/password, the api key always differ.
- **`username`** (*`str`*) – When making a call need HTTP simple authentication username. Mandatory if no referer set
- **`password`** (*`str`*) – When making a call need HTTP simple authentication password. Mandatory if no referer set
- **`referer`** (*`str`*) – When making a call need HTTP referer. Mandatory if no password and username
- **`domain`** (*`str`*) – Currently it is `'wxs.ign.fr'`, can be changed for testing purposes for developer API e.g `'gpp3-wxs.ign.fr'` at the moment.
- **`scheme`** (*`str`*) – See *`geopy.geocoders.options.default_scheme`*.
- **`timeout`** (*`int`*) – See *`geopy.geocoders.options.default_timeout`*.
- **`proxies`** (*`dict`*) – See *`geopy.geocoders.options.default_proxies`*.
- **`user_agent`** (*`str`*) – See *`geopy.geocoders.options.default_user_agent`*.

  New in version 1.12.0.
- **`format_string`** (*`str`*) – See *`geopy.geocoders.options.default_format_string`*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **`ssl_context`** (`ssl.SSLContext`) – See *`geopy.geocoders.options.default_ssl_context`*.

  New in version 1.14.0.

`geocode`(*query*, *query_type='StreetAddress'*, *maximum_responses=25*, *is_freeform=False*, *filtering=None*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **`query`** (*`str`*) – The query string to be geocoded.
- **`query_type`** (*`str`*) – The type to provide for geocoding. It can be *PositionOfInterest*, *StreetAddress* or *CadastralParcel*. *StreetAddress* is the default choice if none provided.
- **`maximum_responses`** (*`int`*) – The maximum number of responses to ask to the API in the query body.
- **`is_freeform`** (*`str`*) – Set if return is structured with freeform structure or a more structured returned. By default, value is False.

- **filtering** (*str*) – Provide string that help setting geocoder filter. It contains an XML string. See examples in documentation and ignfrance.py file in directory tests.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse (*query*, *reverse_geocode_preference=('StreetAddress',)*, *maximum_responses=25*, *filtering=''*, *exactly_one=DEFAULT_SENTINEL*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **reverse_geocode_preference** (*list*) – Enable to set expected results type. It can be *StreetAddress* or *PositionOfInterest*. Default is set to *StreetAddress*.
- **maximum_responses** (*int*) – The maximum number of responses to ask to the API in the query body.
- **filtering** (*str*) – Provide string that help setting geocoder filter. It contains an XML string. See examples in documentation and ignfrance.py file in directory tests.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.14.0: Default value for `exactly_one` was `False`, which differs from the conventional default across geopy. Please always pass this argument explicitly, otherwise you would get a warning. In geopy 2.0 the default value will become `True`.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.21 MapBox

class geopy.geocoders.**MapBox** (*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.mapbox.com'*)

Geocoder using the Mapbox API.

Documentation at: https://www.mapbox.com/api-documentation/

New in version 1.17.0.

__init__ (*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.mapbox.com'*)

Parameters

- **api_key** (*str*) – The API key required by Mapbox to perform geocoding requests. API keys are managed through Mapox's account page (https://www.mapbox.com/account/access-tokens).
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.
- **domain** (*str*) – base api domain for mapbox

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *proximity=None*, *country=None*, *bbox=None*)

Return a location point by address.

Changed in version 1.20.0: Previously due to a bug the resulting *geopy.location.Location*'s `raw` attribute contained a single string instead of a full service response.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.20.0: Previously due to a bug this parameter wasn't respected.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **proximity** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – A coordinate to bias local results based on a provided location.
- **country** (*str or list*) – Country to filter result in form of ISO 3166-1 alpha-2 country code (e.g. `FR`). Might be a Python list of strings.

  Changed in version 1.19.0: Previously only a single string could be specified. Now a Python list of individual countries is supported.
- **bbox** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – The bounding box of the viewport within which to bias geocode results more prominently. Example: `[Point(22, 180), Point(-22, -180)]`.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Changed in version 1.20.0: Previously due to a bug the resulting *geopy.location.Location*'s `raw` attribute contained a single string instead of a full service response.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.22 MapQuest

class geopy.geocoders.**MapQuest**(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='www.mapquestapi.com'*)

Geocoder using the MapQuest API based on Licensed data.

Documentation at: https://developer.mapquest.com/documentation/geocoding-api/

MapQuest provides two Geocoding APIs:

- *geopy.geocoders.OpenMapQuest* Nominatim-alike API which is based on Open data from OpenStreetMap.
- *geopy.geocoders.MapQuest* (this class) MapQuest's own API which is based on Licensed data.

New in version 1.22.0.

__init__(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='www.mapquestapi.com'*)

Parameters

- **api_key** (*str*) – The API key required by Mapquest to perform geocoding requests. API keys are managed through MapQuest's "Manage Keys" page (https://developer.mapquest.com/user/me/apps).
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.
- **domain** (*str*) – base api domain for mapquest

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *bounds=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Limit the maximum number of items in the response. This will be reset to one if `exactly_one` is True.
- **bounds** (list or tuple of 2 items of `geopy.point.Point` or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – The bounding box of the viewport within which to bias geocode results more prominently. Example: `[Point(22, 180), Point(-22, -180)]`.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.23 MapTiler

class `geopy.geocoders.`**MapTiler**(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.maptiler.com'*)

Geocoder using the MapTiler API.

Documentation at: https://cloud.maptiler.com/geocoding/ (requires sign-up)

New in version 1.22.0.

__init__(*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.maptiler.com'*)

Parameters

- **api_key** (*str*) – The API key required by Maptiler to perform geocoding requests. API keys are managed through Maptiler's account page (https://cloud.maptiler.com/account/keys).
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.
- **domain** (*str*) – base api domain for Maptiler

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *proximity=None*, *language=None*, *bbox=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **proximity** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – A coordinate to bias local results based on a provided location.
- **language** (*str or list*) – Prefer results in specific languages. Accepts a single string like `"en"` or a list like `["de", "en"]`.
- **bbox** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – The bounding box of the viewport within which to bias geocode results more prominently. Example: `[Point(22, 180), Point(-22, -180)]`.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=None*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str or list*) – Prefer results in specific languages. Accepts a single string like `"en"` or a list like `["de", "en"]`.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.24 OpenCage

class geopy.geocoders.**OpenCage**(*api_key*, *domain='api.opencagedata.com'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the OpenCageData API.

Documentation at: https://opencagedata.com/api

New in version 1.1.0.

__init__(*api_key*, *domain='api.opencagedata.com'*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – The API key required by OpenCageData to perform geocoding requests. You can get your key here: https://opencagedata.com/
- **domain** (*str*) – Currently it is `'api.opencagedata.com'`, can be changed for testing purposes.
- **scheme** (*str*) – See `geopy.geocoders.options.default_scheme`.
- **timeout** (*int*) – See `geopy.geocoders.options.default_timeout`.
- **proxies** (*dict*) – See `geopy.geocoders.options.default_proxies`.
- **user_agent** (*str*) – See `geopy.geocoders.options.default_user_agent`.

  New in version 1.12.0.
- **format_string** (*str*) – See `geopy.geocoders.options.default_format_string`.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **ssl_context** (`ssl.SSLContext`) – See `geopy.geocoders.options.default_ssl_context`.

  New in version 1.14.0.

geocode(*query*, *bounds=None*, *country=None*, *language=None*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **language** (*str*) – an IETF format language code (such as *es* for Spanish or pt-BR for Brazilian Portuguese); if this is omitted a code of *en* (English) will be assumed by the remote service.
- **bounds** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Provides the geocoder with a hint to the region that the query resides in. This value will help the geocoder but will not restrict the possible results to the supplied region. The bounds parameter should be specified as 2 coordinate points – corners of a bounding box. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.17.0: Previously the only supported format for bounds was a string of `"longitude,latitude,longitude,latitude"`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.
- **country** (*str or list*) – Restricts the results to the specified country or countries. The country code is a 2 character code as defined by the ISO 3166-1 Alpha 2 standard (e.g. `fr`). Might be a Python list of strings.

  Changed in version 1.19.0: This parameter didn't seem to be respected previously. Also, previously only a single string could be specified. Now a Python list of individual countries is supported.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *language=None*, *exactly_one=DEFAULT_SENTINEL*, *timeout=DEFAULT_SENTINEL*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **language** (*str*) – The language in which to return results.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.14.0: Default value for `exactly_one` was `False`, which differs from the conventional default across geopy. Please always pass this argument explicitly, otherwise you would get a warning. In geopy 2.0 the default value will become `True`.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.25 OpenMapQuest

class geopy.geocoders.**OpenMapQuest**(*api_key=None*, *format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='open.mapquestapi.com'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Bases: `geopy.geocoders.osm.Nominatim`

Geocoder using MapQuest Open Platform Web Services.

Documentation at: https://developer.mapquest.com/documentation/open/

MapQuest provides two Geocoding APIs:

- *geopy.geocoders.OpenMapQuest* (this class) Nominatim-alike API which is based on Open data from OpenStreetMap.
- *geopy.geocoders.MapQuest* MapQuest's own API which is based on Licensed data.

Changed in version 1.17.0: OpenMapQuest now extends the Nominatim class.

__init__(*api_key=None*, *format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='open.mapquestapi.com'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – API key provided by MapQuest, required.

  Changed in version 1.12.0: OpenMapQuest now requires an API key. Using an empty key will result in a *geopy.exc.ConfigurationError*.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.

- **view_box** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.17.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *viewbox* instead.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box.

  New in version 1.17.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *bounded* instead.

- **country_bias** (*str or list*) – Limit search results to a specific country. This param sets a default value for the *geocode*'s `country_codes`.

  New in version 1.17.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_codes* instead.

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **domain** (*str*) – Domain where the target Nominatim service is hosted.

  New in version 1.17.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *addressdetails=False*, *language=False*, *geometry=None*, *extratags=False*, *country_codes=None*, *viewbox=None*, *bounded=None*, *featuretype=None*, *namedetails=False*)

Return a location point by address.

Parameters

- **query** (*dict or str*) – The address, query or a structured query you wish to geocode.

  Changed in version 1.0.0: For a structured query, provide a dictionary whose keys are one of: *street*, *city*, *county*, *state*, *country*, or *postalcode*. For more information, see Nominatim's documentation for *structured requests*:

  > https://nominatim.org/release-docs/develop/api/Search
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Maximum amount of results to return from Nominatim. Unless exactly_one is set to False, limit will always be 1.

  New in version 1.13.0.
- **addressdetails** (*bool*) – If you want in *Location.raw* to include addressdetails such as city_district, etc set it to True
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.
- **geometry** (*str*) – If present, specifies whether the geocoding service should return the result's geometry in *wkt*, *svg*, *kml*, or *geojson* formats. This is available via the *raw* attribute on the returned *geopy.location.Location* object.

  New in version 1.3.0.
- **extratags** (*bool*) – Include additional information in the result if available, e.g. wikipedia link, opening hours.

  New in version 1.17.0.

- **country_codes** (*str or list*) – Limit search results to a specific country (or a list of countries). A country_code should be the ISO 3166-1alpha2 code, e.g. `gb` for the United Kingdom, `de` for Germany, etc.

  New in version 1.19.0.

- **viewbox** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.19.0.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box. Defaults to *False*.

  New in version 1.19.0.

- **featuretype** (*str*) – If present, restrict results to certain type of features. Allowed values: *country*, *state*, *city*, *settlement*.

  New in version 1.21.0.

- **namedetails** (*bool*) – If you want in *Location.raw* to include namedetails, set it to True. This will be a list of alternative names, including language variants, etc.

  New in version 1.21.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=False*, *addressdetails=True*, *zoom=None*)
Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.

- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.

- **addressdetails** (*bool*) – Whether or not to include address details, such as city, county, state, etc. in *Location.raw*

  New in version 1.14.0.

- **zoom** (*int*) – Level of detail required for the address, an integer in range from 0 (country level) to 18 (building level), default is 18.

  New in version 1.22.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.26 Nominatim

class geopy.geocoders.**Nominatim**(*format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='nominatim.openstreetmap.org'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Nominatim geocoder for OpenStreetMap data.

Documentation at: https://nominatim.org/release-docs/develop/api/Overview/

> **Attention:** Using Nominatim with the default *user_agent* is strongly discouraged, as it violates Nominatim's Usage Policy https://operations.osmfoundation.org/policies/nominatim/ and may possibly cause 403 and 429 HTTP errors. Please make sure to specify a custom *user_agent* with `Nominatim(user_agent="my-application")` or by overriding the default *user_agent*: `geopy.geocoders.options.default_user_agent = "my-application"`. In geopy 2.0 an exception will be thrown when a custom *user_agent* is not specified.

Changed in version 1.16.0: A warning is now issued when a default *user_agent* is used which restates the *Attention* block above.

__init__(*format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='nominatim.openstreetmap.org'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.

- **view_box** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.15.0: Previously only a list of stringified coordinates was supported.

  Changed in version 1.17.0: Previously view_box could be a list of 4 strings or numbers in the format of `[longitude, latitude, longitude, latitude]`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *viewbox* instead.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box.

  New in version 1.15.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *bounded* instead.

- **country_bias** (*str or list*) – Limit search results to a specific country. This param sets a default value for the *geocode*'s `country_codes`.

Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_codes* instead.

- **timeout** (*int*) – See `geopy.geocoders.options.default_timeout`.
- **proxies** (*dict*) – See `geopy.geocoders.options.default_proxies`.
- **domain** (*str*) – Domain where the target Nominatim service is hosted.

  New in version 1.8.2.
- **scheme** (*str*) – See `geopy.geocoders.options.default_scheme`.

  New in version 1.8.2.
- **user_agent** (*str*) – See `geopy.geocoders.options.default_user_agent`.

  New in version 1.12.0.
- **ssl_context** (`ssl.SSLContext`) – See `geopy.geocoders.options.default_ssl_context`.

  New in version 1.14.0.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *addressdetails=False*, *language=False*, *geometry=None*, *extratags=False*, *country_codes=None*, *viewbox=None*, *bounded=None*, *featuretype=None*, *namedetails=False*)

Return a location point by address.

Parameters

- **query** (*dict or str*) – The address, query or a structured query you wish to geocode.

  Changed in version 1.0.0: For a structured query, provide a dictionary whose keys are one of: *street*, *city*, *county*, *state*, *country*, or *postalcode*. For more information, see Nominatim's documentation for *structured requests*:

  https://nominatim.org/release-docs/develop/api/Search
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Maximum amount of results to return from Nominatim. Unless exactly_one is set to False, limit will always be 1.

  New in version 1.13.0.
- **addressdetails** (*bool*) – If you want in *Location.raw* to include addressdetails such as city_district, etc set it to True
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.
- **geometry** (*str*) – If present, specifies whether the geocoding service should return the result's geometry in *wkt*, *svg*, *kml*, or *geojson* formats. This is available via the *raw* attribute on the returned `geopy.location.Location` object.

  New in version 1.3.0.

- **extratags** (*bool*) – Include additional information in the result if available, e.g. wikipedia link, opening hours.

  New in version 1.17.0.

- **country_codes** (*str or list*) – Limit search results to a specific country (or a list of countries). A country_code should be the ISO 3166-1alpha2 code, e.g. `gb` for the United Kingdom, `de` for Germany, etc.

  New in version 1.19.0.

- **viewbox** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.19.0.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box. Defaults to *False*.

  New in version 1.19.0.

- **featuretype** (*str*) – If present, restrict results to certain type of features. Allowed values: *country*, *state*, *city*, *settlement*.

  New in version 1.21.0.

- **namedetails** (*bool*) – If you want in *Location.raw* to include namedetails, set it to True. This will be a list of alternative names, including language variants, etc.

  New in version 1.21.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=False*, *addressdetails=True*, *zoom=None*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.

- **exactly_one** (*bool*) – Return one result or a list of results, if available.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.

- **addressdetails** (*bool*) – Whether or not to include address details, such as city, county, state, etc. in *Location.raw*

  New in version 1.14.0.

- **zoom** (*int*) – Level of detail required for the address, an integer in range from 0 (country level) to 18 (building level), default is 18.

  New in version 1.22.0.

Return type `None`, *`geopy.location.Location`* or a list of them, if `exactly_one=False`.

3.27 Pelias

class `geopy.geocoders.`**`Pelias`**(*domain*, *api_key=None*, *format_string=None*, *boundary_rect=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *scheme=None*, *ssl_context=DEFAULT_SENTINEL*)

Pelias geocoder.

Documentation at: https://github.com/pelias/documentation

See also *`geopy.geocoders.GeocodeEarth`* which is a Pelias-based service provided by the developers of Pelias itself.

Changed in version 1.15.0: `Mapzen` geocoder has been renamed to `Pelias`.

`__init__`(*domain*, *api_key=None*, *format_string=None*, *boundary_rect=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *scheme=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **`domain`** (*`str`*) – Specify a domain for Pelias API.
- **`api_key`** (*`str`*) – Pelias API key, optional.
- **`format_string`** (*`str`*) – See *`geopy.geocoders.options.default_format_string`*.

  Deprecated since version 1.22.0.

- **`boundary_rect`** (list or tuple of 2 items of *`geopy.point.Point`* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.17.0: Previously boundary_rect could be a list of 4 strings or numbers in the format of `[longitude, latitude, longitude, latitude]`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *boundary_rect* instead.

- **`country_bias`** (*`str`*) – Bias results to this country (ISO alpha-3).

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_bias* instead.

- **`timeout`** (*`int`*) – See *`geopy.geocoders.options.default_timeout`*.
- **`proxies`** (*`dict`*) – See *`geopy.geocoders.options.default_proxies`*.
- **`user_agent`** (*`str`*) – See *`geopy.geocoders.options.default_user_agent`*.
- **`scheme`** (*`str`*) – See *`geopy.geocoders.options.default_scheme`*.
- **`ssl_context`** (`ssl.SSLContext`) – See *`geopy.geocoders.options.default_ssl_context`*.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *boundary_rect=None*, *country_bias=None*, *language=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address, query or structured query to geocode you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **boundary_rect** (list or tuple of 2 items of `geopy.point.Point` or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.19.0.

- **country_bias** (*str*) – Bias results to this country (ISO alpha-3).

  New in version 1.19.0.

- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.21.0.

Return type None, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=None*)

Return an address by location point.

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.21.0.

Return type None, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.28 Photon

class geopy.geocoders.**Photon**(*format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='photon.komoot.de'*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using Photon geocoding service (data based on OpenStreetMap and service provided by Komoot on https://photon.komoot.de).

Documentation at: https://github.com/komoot/photon

Photon/Komoot geocoder aims to let you *search as you type with OpenStreetMap*. No API Key is needed by this platform.

`__init__`(*format_string=None, scheme=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, domain='photon.komoot.de', user_agent=None, ssl_context=DEFAULT_SENTINEL*)

Parameters

- **`format_string`** (`str`) – See `geopy.geocoders.options.default_format_string`.

  Deprecated since version 1.22.0.

- **`scheme`** (`str`) – See `geopy.geocoders.options.default_scheme`.
- **`timeout`** (`int`) – See `geopy.geocoders.options.default_timeout`.
- **`proxies`** (`dict`) – See `geopy.geocoders.options.default_proxies`.
- **`domain`** (`str`) – Should be the localized Photon domain to connect to. The default is `'photon.komoot.de'`, but you can change it to a domain of your own.
- **`user_agent`** (`str`) – See `geopy.geocoders.options.default_user_agent`.

  New in version 1.12.0.

- **`ssl_context`** (`ssl.SSLContext`) – See `geopy.geocoders.options.default_ssl_context`.

  New in version 1.14.0.

`geocode`(*query, exactly_one=True, timeout=DEFAULT_SENTINEL, location_bias=None, language=False, limit=None, osm_tag=None*)

Return a location point by address.

Parameters

- **`query`** (`str`) – The address or query you wish to geocode.
- **`exactly_one`** (`bool`) – Return one result or a list of results, if available.
- **`timeout`** (`int`) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **`location_bias`** – The coordinates to used as location bias.
- **`language`** (`str`) – Preferred language in which to return results.
- **`limit`** (`int`) – Limit the number of returned results, defaults to no limit.

  New in version 1.12.0.

- **`osm_tag`** (`str or list or set`) – The expression to filter (include/exclude) by key and/ or value, str as `'key:value'` or list/set of str if multiple filters are required as `['key:!val', '!key', ':!value']`.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=False*, *limit=None*)
Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Preferred language in which to return results.
- **limit** (*int*) – Limit the number of returned results, defaults to no limit.

  New in version 1.12.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.29 PickPoint

class geopy.geocoders.**PickPoint** (*api_key*, *format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='api.pickpoint.io'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)
Bases: `geopy.geocoders.osm.Nominatim`

PickPoint geocoder is a commercial version of Nominatim.

Documentation at: https://pickpoint.io/api-reference

New in version 1.13.0.

__init__ (*api_key*, *format_string=None*, *view_box=None*, *bounded=None*, *country_bias=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *domain='api.pickpoint.io'*, *scheme=None*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – PickPoint API key obtained at https://pickpoint.io.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **view_box** (list or tuple of 2 items of *geopy.point.Point* or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  Changed in version 1.17.0: Previously view_box could be a list of 4 strings or numbers in the format of `[longitude, latitude, longitude, latitude]`. This format is now deprecated in favor of a list/tuple of a pair of geopy Points and will be removed in geopy 2.0.

Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *viewbox* instead.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *bounded* instead.

- **country_bias** (*str or list*) – Limit search results to a specific country. This param sets a default value for the *geocode*'s `country_codes`.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *country_codes* instead.

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **domain** (*str*) – Domain where the target Nominatim service is hosted.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *addressdetails=False*, *language=False*, *geometry=None*, *extratags=False*, *country_codes=None*, *viewbox=None*, *bounded=None*, *featuretype=None*, *namedetails=False*)

Return a location point by address.

Parameters

- **query** (*dict or str*) – The address, query or a structured query you wish to geocode.

  Changed in version 1.0.0: For a structured query, provide a dictionary whose keys are one of: *street*, *city*, *county*, *state*, *country*, or *postalcode*. For more information, see Nominatim's documentation for *structured requests*:

  https://nominatim.org/release-docs/develop/api/Search

- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Maximum amount of results to return from Nominatim. Unless exactly_one is set to False, limit will always be 1.

  New in version 1.13.0.

- **addressdetails** (*bool*) – If you want in *Location.raw* to include addressdetails such as city_district, etc set it to True
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.

- **geometry** (*str*) – If present, specifies whether the geocoding service should return the result's geometry in *wkt*, *svg*, *kml*, or *geojson* formats. This is available via the *raw* attribute on the returned `geopy.location.Location` object.

  New in version 1.3.0.

- **extratags** (*bool*) – Include additional information in the result if available, e.g. wikipedia link, opening hours.

  New in version 1.17.0.

- **country_codes** (*str or list*) – Limit search results to a specific country (or a list of countries). A country_code should be the ISO 3166-1alpha2 code, e.g. `gb` for the United Kingdom, `de` for Germany, etc.

  New in version 1.19.0.

- **viewbox** (list or tuple of 2 items of `geopy.point.Point` or `(latitude, longitude)` or `"%(latitude)s, %(longitude)s"`.) – Coordinates to restrict search within. Example: `[Point(22, 180), Point(-22, -180)]`.

  New in version 1.19.0.

- **bounded** (*bool*) – Restrict the results to only items contained within the bounding view_box. Defaults to *False*.

  New in version 1.19.0.

- **featuretype** (*str*) – If present, restrict results to certain type of features. Allowed values: *country*, *state*, *city*, *settlement*.

  New in version 1.21.0.

- **namedetails** (*bool*) – If you want in *Location.raw* to include namedetails, set it to True. This will be a list of alternative names, including language variants, etc.

  New in version 1.21.0.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

reverse(*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=False*, *addressdetails=True*, *zoom=None*)

Return an address by location point.

Parameters

- **query** (`geopy.point.Point`, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Preferred language in which to return results. Either uses standard RFC2616 accept-language string or a simple comma-separated list of language codes.

  New in version 1.0.0.

- **addressdetails** (*bool*) – Whether or not to include address details, such as city, county, state, etc. in *Location.raw*

  New in version 1.14.0.

- **zoom** (*int*) – Level of detail required for the address, an integer in range from 0 (country level) to 18 (building level), default is 18.

  New in version 1.22.0.

Return type None, *geopy.location.Location* or a list of them, if exactly_one=False.

3.30 LiveAddress

class geopy.geocoders.**LiveAddress**(*auth_id*, *auth_token*, *candidates=None*, *scheme='https'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Geocoder using the LiveAddress API provided by SmartyStreets.

Documentation at: https://smartystreets.com/docs/cloud/us-street-api

__init__(*auth_id*, *auth_token*, *candidates=None*, *scheme='https'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *format_string=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **auth_id** (*str*) – Valid *Auth ID* from SmartyStreets.

  New in version 1.5.0.

- **auth_token** (*str*) – Valid *Auth Token* from SmartyStreets.
- **candidates** (*int*) – An integer between 1 and 10 indicating the max number of candidate addresses to return if a valid address could be found. Defaults to *1*.

  Deprecated since version 1.19.0: This argument will be removed in geopy 2.0. Use *geocode*'s *candidates* instead.

- **scheme** (*str*) – Must be https.

  Deprecated since version 1.14.0: Don't use this parameter, it's going to be removed in geopy 2.0.

  Changed in version 1.8.0: LiveAddress now requires *https*. Specifying *scheme=http* will result in a *geopy.exc.ConfigurationError*.

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.

- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.

- **ssl_context** (*ssl.SSLContext*) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *candidates=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a `geopy.exc.GeocoderTimedOut` exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **candidates** (*int*) – An integer between 1 and 10 indicating the max number of candidate addresses to return if a valid address could be found. Defaults to *1*.

  New in version 1.19.0.

Return type `None`, `geopy.location.Location` or a list of them, if `exactly_one=False`.

3.31 TomTom

class geopy.geocoders.**TomTom** (*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.tomtom.com'*)

TomTom geocoder.

Documentation at: https://developer.tomtom.com/search-api/search-api-documentation

New in version 1.15.0.

__init__ (*api_key*, *format_string=None*, *scheme=None*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*, *domain='api.tomtom.com'*)

Parameters

- **api_key** (*str*) – TomTom API key.
- **format_string** (*str*) – See `geopy.geocoders.options.default_format_string`.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – See `geopy.geocoders.options.default_scheme`.
- **timeout** (*int*) – See `geopy.geocoders.options.default_timeout`.
- **proxies** (*dict*) – See `geopy.geocoders.options.default_proxies`.
- **user_agent** (*str*) – See `geopy.geocoders.options.default_user_agent`.
- **ssl_context** (`ssl.SSLContext`) – See `geopy.geocoders.options.default_ssl_context`.
- **domain** (*str*) – Domain where the target TomTom service is hosted.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *limit=None*, *typeahead=False*, *language=None*)

Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **limit** (*int*) – Maximum amount of results to return from the service. Unless exactly_one is set to False, limit will always be 1.
- **typeahead** (*bool*) – If the "typeahead" flag is set, the query will be interpreted as a partial input and the search will enter predictive mode.
- **language** (*str*) – Language in which search results should be returned. When data in specified language is not available for a specific field, default language is used. List of supported languages (case-insensitive): https://developer.tomtom.com/online-search/online-search-documentation/supported-languages

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *language=None*)

Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **language** (*str*) – Language in which search results should be returned. When data in specified language is not available for a specific field, default language is used. List of supported languages (case-insensitive): https://developer.tomtom.com/online-search/online-search-documentation/supported-languages

  New in version 1.18.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.32 What3Words

class `geopy.geocoders.`**What3Words** (*api_key*, *format_string=None*, *scheme='https'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

What3Words geocoder.

Documentation at: https://docs.what3words.com/api/v2/

New in version 1.5.0.

Changed in version 1.15.0: API has been updated to v2.

__init__(*api_key*, *format_string=None*, *scheme='https'*, *timeout=DEFAULT_SENTINEL*, *proxies=DEFAULT_SENTINEL*, *user_agent=None*, *ssl_context=DEFAULT_SENTINEL*)

Parameters

- **api_key** (*str*) – Key provided by What3Words (https://accounts.what3words.com/register).
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  Deprecated since version 1.22.0.
- **scheme** (*str*) – Must be `https`.

  Deprecated since version 1.15.0: API v2 requires https. Don't use this parameter, it's going to be removed in geopy 2.0. Scheme other than `https` would result in a *geopy.exc.ConfigurationError* being thrown.
- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **ssl_context** (`ssl.SSLContext`) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode(*query*, *lang='en'*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a location point for a *3 words* query. If the *3 words* address doesn't exist, a *geopy.exc.GeocoderQueryError* exception will be thrown.

Parameters

- **query** (*str*) – The 3-word address you wish to geocode.
- **lang** (*str*) – two character language codes as supported by the API (https://docs.what3words.com/api/v2/#lang).
- **exactly_one** (*bool*) – Return one result or a list of results, if available. Due to the address scheme there is always exactly one result for each *3 words* address, so this parameter is rather useless for this geocoder.

  Changed in version 1.14.0: `exactly_one=False` now returns a list of a single location. This option wasn't respected before.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse(*query*, *lang='en'*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*)

Return a *3 words* address by location point. Each point on surface has a *3 words* address, so there's always a non-empty response.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the 3 word address.

- **lang** (*str*) – two character language codes as supported by the API (https://docs.what3words.com/api/v2/#lang).
- **exactly_one** (*bool*) – Return one result or a list of results, if available. Due to the address scheme there is always exactly one result for each *3 words* address, so this parameter is rather useless for this geocoder.

  Changed in version 1.14.0: `exactly_one=False` now returns a list of a single location. This option wasn't respected before.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

Return type *geopy.location.Location* or a list of them, if `exactly_one=False`.

3.33 Yandex

class geopy.geocoders.**Yandex**(*api_key=None, lang=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, scheme=None, format_string=None, ssl_context=DEFAULT_SENTINEL*)

Yandex geocoder.

Documentation at: https://tech.yandex.com/maps/doc/geocoder/desc/concepts/input_params-docpage/

New in version 1.5.0.

> **Attention:** Since September 2019 Yandex requires each request to have an API key. API keys can be created at https://developer.tech.yandex.ru/

__init__(*api_key=None, lang=None, timeout=DEFAULT_SENTINEL, proxies=DEFAULT_SENTINEL, user_agent=None, scheme=None, format_string=None, ssl_context=DEFAULT_SENTINEL*)

Changed in version 1.14.0: Default scheme has been changed from `http` to `https`.

Parameters

- **api_key** (*str*) – Yandex API key, mandatory. The key can be created at https://developer.tech.yandex.ru/

  Changed in version 1.21.0: API key is mandatory since September 2019.
- **lang** (*str*) – Language of the response and regional settings of the map. List of supported values:
  - `tr_TR` – Turkish (only for maps of Turkey);
  - `en_RU` – response in English, Russian map features;
  - `en_US` – response in English, American map features;
  - `ru_RU` – Russian (default);
  - `uk_UA` – Ukrainian;
  - `be_BY` – Belarusian.

  Deprecated since version 1.22.0: This argument will be removed in geopy 2.0. Use *geocode*'s and *reverse*'s *lang* instead.

- **timeout** (*int*) – See *geopy.geocoders.options.default_timeout*.
- **proxies** (*dict*) – See *geopy.geocoders.options.default_proxies*.
- **user_agent** (*str*) – See *geopy.geocoders.options.default_user_agent*.

  New in version 1.12.0.
- **scheme** (*str*) – See *geopy.geocoders.options.default_scheme*.

  New in version 1.14.0.
- **format_string** (*str*) – See *geopy.geocoders.options.default_format_string*.

  New in version 1.14.0.

  Deprecated since version 1.22.0.
- **ssl_context** (ssl.SSLContext) – See *geopy.geocoders.options.default_ssl_context*.

  New in version 1.14.0.

geocode (*query*, *exactly_one=True*, *timeout=DEFAULT_SENTINEL*, *lang=None*)
Return a location point by address.

Parameters

- **query** (*str*) – The address or query you wish to geocode.
- **exactly_one** (*bool*) – Return one result or a list of results, if available.
- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.
- **lang** (*str*) – Language of the response and regional settings of the map. List of supported values:
  - `tr_TR` – Turkish (only for maps of Turkey);
  - `en_RU` – response in English, Russian map features;
  - `en_US` – response in English, American map features;
  - `ru_RU` – Russian (default);
  - `uk_UA` – Ukrainian;
  - `be_BY` – Belarusian.

  New in version 1.22.0.

Return type None, *geopy.location.Location* or a list of them, if `exactly_one=False`.

reverse (*query*, *exactly_one=DEFAULT_SENTINEL*, *timeout=DEFAULT_SENTINEL*, *kind=None*, *lang=None*)
Return an address by location point.

Parameters

- **query** (*geopy.point.Point*, list or tuple of `(latitude, longitude)`, or string as `"%(latitude)s, %(longitude)s"`.) – The coordinates for which you wish to obtain the closest human-readable addresses.

- **exactly_one** (*bool*) – Return one result or a list of results, if available.

  Changed in version 1.14.0: Default value for `exactly_one` was `False`, which differs from the conventional default across geopy. Please always pass this argument explicitly, otherwise you would get a warning. In geopy 2.0 the default value will become `True`.

- **timeout** (*int*) – Time, in seconds, to wait for the geocoding service to respond before raising a *geopy.exc.GeocoderTimedOut* exception. Set this only if you wish to override, on this call only, the value set during the geocoder's initialization.

- **kind** (*str*) – Type of toponym. Allowed values: *house*, *street*, *metro*, *district*, *locality*.

  New in version 1.14.0.

- **lang** (*str*) – Language of the response and regional settings of the map. List of supported values:

  - `tr_TR` – Turkish (only for maps of Turkey);
  - `en_RU` – response in English, Russian map features;
  - `en_US` – response in English, American map features;
  - `ru_RU` – Russian (default);
  - `uk_UA` – Ukrainian;
  - `be_BY` – Belarusian.

  New in version 1.22.0.

Return type `None`, *geopy.location.Location* or a list of them, if `exactly_one=False`.

CHAPTER 4

Calculating Distance

Geopy can calculate geodesic distance between two points using the geodesic distance or the great-circle distance, with a default of the geodesic distance available as the function `geopy.distance.distance`.

Great-circle distance (*great_circle*) uses a spherical model of the earth, using the mean earth radius as defined by the International Union of Geodesy and Geophysics, $(2a + b)/3$ = 6371.0087714150598 kilometers approx 6371.009 km (for WGS-84), resulting in an error of up to about 0.5%. The radius value is stored in `distance.EARTH_RADIUS`, so it can be customized (it should always be in kilometers, however).

The geodesic distance is the shortest distance on the surface of an ellipsoidal model of the earth. The default algorithm uses the method is given by Karney (2013) (*geodesic*); this is accurate to round-off and always converges. An older *deprecated* method due to Vincenty (1975) (*vincenty*) is also available; this is only accurate to 0.2 mm and the distance calculation fails to converge for nearly antipodal points.

`geopy.distance.distance` currently uses *geodesic*.

There are multiple popular ellipsoidal models, and which one will be the most accurate depends on where your points are located on the earth. The default is the WGS-84 ellipsoid, which is the most globally accurate. geopy includes a few other models in the `distance.ELLIPSOIDS` dictionary:

```
              model             major (km)   minor (km)     flattening
ELLIPSOIDS = {'WGS-84':        (6378.137,    6356.7523142,  1 / 298.257223563),
              'GRS-80':        (6378.137,    6356.7523141,  1 / 298.257222101),
              'Airy (1830)':   (6377.563396, 6356.256909,   1 / 299.3249646),
              'Intl 1924':     (6378.388,    6356.911946,   1 / 297.0),
              'Clarke (1880)': (6378.249145, 6356.51486955, 1 / 293.465),
              'GRS-67':        (6378.1600,   6356.774719,   1 / 298.25),
              }
```

Here are examples of `distance.distance` usage:

```
>>> from geopy import distance
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(distance.distance(newport_ri, cleveland_oh).miles)
538.39044536
```

(continues on next page)

(continued from previous page)

```
>>> wellington = (-41.32, 174.81)
>>> salamanca = (40.96, -5.50)
>>> print(distance.distance(wellington, salamanca).km)
19959.6792674
```

The second example above fails with *vincenty*.

Using *great_circle* distance:

```
>>> print(distance.great_circle(newport_ri, cleveland_oh).miles)
536.997990696
```

You can change the ellipsoid model used by the geodesic formulas like so:

```
>>> ne, cl = newport_ri, cleveland_oh
>>> print(distance.geodesic(ne, cl, ellipsoid='GRS-80').miles)
```

The above model name will automatically be retrieved from the `distance.ELLIPSOIDS` dictionary. Alternatively, you can specify the model values directly:

```
>>> distance.geodesic(ne, cl, ellipsoid=(6377., 6356., 1 / 297.)).miles
```

Distances support simple arithmetic, making it easy to do things like calculate the length of a path:

```
>>> from geopy import Nominatim
>>> d = distance.distance
>>> g = Nominatim(user_agent="specify_your_app_name_here")
>>> _, wa = g.geocode('Washington, DC')
>>> _, pa = g.geocode('Palo Alto, CA')
>>> print((d(ne, cl) + d(cl, wa) + d(wa, pa)).miles)
3277.30439191
```

Currently all algorithms assume that altitudes of the points are either zero (as in the examples above) or equal, and are relatively small. Thus altitudes never affect the resulting distances:

```
>>> from geopy import distance
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(distance.distance(newport_ri, cleveland_oh).km)
866.4554329098687
>>> newport_ri = (41.49008, -71.312796, 100)
>>> cleveland_oh = (41.499498, -81.695391, 100)
>>> print(distance.distance(newport_ri, cleveland_oh).km)
866.4554329098687
```

If you need to calculate distances with elevation, then for short distances the Euclidean distance formula might give a suitable approximation:

```
>>> import math
>>> from geopy import distance
>>> p1 = (43.668613, 40.258916, 0.976)
>>> p2 = (43.658852, 40.250839, 1.475)
>>> flat_distance = distance.distance(p1[:2], p2[:2]).km
>>> print(flat_distance)
1.265133525952866
```

(continues on next page)

(continued from previous page)

```
>>> euclidian_distance = math.sqrt(flat_distance**2 + (p2[2] - p1[2])**2)
>>> print(euclidian_distance)
1.359986705262199
```

Changed in version 1.23.0: Calculating distances between points with different altitudes now causes a deprecation warning. In geopy 2.0 this will become a `ValueError` exception.

`geopy.distance.`**`lonlat`**`(`*x*, *y*, *z=0*`)`

`geopy.distance.distance` accepts coordinates in `(y, x)/(lat, lon)` order, while some other libraries and systems might use `(x, y)/(lon, lat)`.

This function provides a convenient way to convert coordinates of the `(x, y)/(lon, lat)` format to a *`geopy.point.Point`* instance.

Example:

```
>>> from geopy.distance import lonlat, distance
>>> newport_ri_xy = (-71.312796, 41.49008)
>>> cleveland_oh_xy = (-81.695391, 41.499498)
>>> print(distance(lonlat(*newport_ri_xy), lonlat(*cleveland_oh_xy)).miles)
538.3904453677203
```

Parameters

- **x** – longitude
- **y** – latitude
- **z** – (optional) altitude

Returns Point(latitude, longitude, altitude)

`class` `geopy.distance.`**`geodesic`**`(`**args*, ***kwargs*`)`

Calculate the geodesic distance between two points.

Set which ellipsoidal model of the earth to use by specifying an `ellipsoid` keyword argument. The default is ‘WGS-84’, which is the most globally accurate model. If `ellipsoid` is a string, it is looked up in the *ELLIPSOIDS* dictionary to obtain the major and minor semiaxes and the flattening. Otherwise, it should be a tuple with those values. See the comments above the *ELLIPSOIDS* dictionary for more information.

Example:

```
>>> from geopy.distance import geodesic
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(geodesic(newport_ri, cleveland_oh).miles)
538.390445368
```

New in version 1.13.0.

`__init__``(`**args*, ***kwargs*`)`

Initialize self. See help(type(self)) for accurate signature.

`class` `geopy.distance.`**`vincenty`**`(`**args*, ***kwargs*`)`

Deprecated since version 1.13: Use *`geodesic`* instead. Vincenty will be removed in geopy 2.0.

Calculate the geodesic distance between two points using the Vincenty’s method.

Set which ellipsoidal model of the earth to use by specifying an `ellipsoid` keyword argument. The default is ‘WGS-84’, which is the most globally accurate model. If `ellipsoid` is a string, it is looked up in the

ELLIPSOIDS dictionary to obtain the major and minor semiaxes and the flattening. Otherwise, it should be a tuple with those values. See the comments above the *ELLIPSOIDS* dictionary for more information.

Example:

```
>>> from geopy.distance import vincenty
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(vincenty(newport_ri, cleveland_oh).miles)
538.390445362
```

Note: Vincenty's method for distance fails to converge for some valid (nearly antipodal) points. In such cases, use `geodesic` which always produces an accurate result.

`__init__`(**args*, ***kwargs*)
Initialize self. See help(type(self)) for accurate signature.

class geopy.distance.**great_circle**(**args*, ***kwargs*)
Use spherical geometry to calculate the surface distance between two points.

Set which radius of the earth to use by specifying a `radius` keyword argument. It must be in kilometers. The default is to use the module constant *EARTH_RADIUS*, which uses the average great-circle radius.

Example:

```
>>> from geopy.distance import great_circle
>>> newport_ri = (41.49008, -71.312796)
>>> cleveland_oh = (41.499498, -81.695391)
>>> print(great_circle(newport_ri, cleveland_oh).miles)
536.997990696
```

`__init__`(**args*, ***kwargs*)
Initialize self. See help(type(self)) for accurate signature.

CHAPTER 5

Data

class geopy.location.**Location**(*address=”*, *point=None*, *raw=None*)

Contains a parsed geocoder response. Can be iterated over as `(location<String>, (latitude<float>, longitude<Float))`. Or one can access the properties `address`, `latitude`, `longitude`, or `raw`. The last is a dictionary of the geocoder’s response for this item.

address

Location as a formatted string returned by the geocoder or constructed by geopy, depending on the service.

Return type unicode

altitude

Location’s altitude.

Note: Geocoding services usually don’t consider altitude neither in requests nor in responses, so almost always the value of this property would be zero.

Return type float or None

latitude

Location’s latitude.

Return type float or None

longitude

Location’s longitude.

Return type float or None

point

geopy.point.Point instance representing the location’s latitude, longitude, and altitude.

Return type *geopy.point.Point* or None

raw

Location’s raw, unparsed geocoder response. For details on this, consult the service’s documentation.

Return type dict or None

class geopy.point.**Point**

A geodetic point with latitude, longitude, and altitude.

Latitude and longitude are floating point values in degrees. Altitude is a floating point value in kilometers. The reference level is never considered and is thus application dependent, so be consistent! The default for all values is 0.

Points can be created in a number of ways…

With latitude, longitude, and altitude:

```
>>> p1 = Point(41.5, -81, 0)
>>> p2 = Point(latitude=41.5, longitude=-81)
```

With a sequence of 2 to 3 values (latitude, longitude, altitude):

```
>>> p1 = Point([41.5, -81, 0])
>>> p2 = Point((41.5, -81))
```

Copy another *Point* instance:

```
>>> p2 = Point(p1)
>>> p2 == p1
True
>>> p2 is p1
False
```

Give a string containing at least latitude and longitude:

```
>>> p1 = Point('41.5,-81.0')
>>> p2 = Point('41.5 N -81.0 W')
>>> p3 = Point('-41.5 S, 81.0 E, 2.5km')
>>> p4 = Point('23 26m 22s N 23 27m 30s E 21.0mi')
>>> p5 = Point('''3 26' 22" N 23 27' 30" E''')
```

Point values can be accessed by name or by index:

```
>>> p = Point(41.5, -81.0, 0)
>>> p.latitude == p[0]
True
>>> p.longitude == p[1]
True
>>> p.altitude == p[2]
True
```

When unpacking (or iterating), a `(latitude, longitude, altitude)` tuple is returned:

```
>>> latitude, longitude, altitude = p
```

Textual representations:

```
>>> p = Point(41.5, -81.0, 12.3)
>>> str(p)  # same as `p.format()`
'41 30m 0s N, 81 0m 0s W, 12.3km'
>>> p.format_unicode()
'41° 30 0 N, 81° 0 0 W, 12.3km'
>>> repr(p)
```

(continues on next page)

(continued from previous page)

```
'Point(41.5, -81.0, 12.3)'
>>> repr(tuple(p))
'(41.5, -81.0, 12.3)'
```

static __new__(*cls*, *latitude=None*, *longitude=None*, *altitude=None*)

Parameters

- **latitude** (*float*) – Latitude of point.
- **longitude** (*float*) – Longitude of point.
- **altitude** (*float*) – Altitude of point.

format(*altitude=None*, *deg_char=”*, *min_char=’m’*, *sec_char=’s’*)

Format decimal degrees (DD) to degrees minutes seconds (DMS):

```
>>> p = Point(41.5, -81.0, 12.3)
>>> p.format()
'41 30m 0s N, 81 0m 0s W, 12.3km'
>>> p = Point(41.5, 0, 0)
>>> p.format()
'41 30m 0s N, 0 0m 0s E'
```

See also *format_unicode()*.

Parameters altitude (*bool*) – Whether to include `altitude` value. By default it is automatically included if it is non-zero.

format_altitude(*unit=’km’*)

Format altitude with unit:

```
>>> p = Point(41.5, -81.0, 12.3)
>>> p.format_altitude()
'12.3km'
>>> p = Point(41.5, -81.0, 0)
>>> p.format_altitude()
'0.0km'
```

Parameters unit (*str*) – Resulting altitude unit. Supported units are listed in *from_string()* doc.

format_decimal(*altitude=None*)

Format decimal degrees with altitude:

```
>>> p = Point(41.5, -81.0, 12.3)
>>> p.format_decimal()
'41.5, -81.0, 12.3km'
>>> p = Point(41.5, 0, 0)
>>> p.format_decimal()
'41.5, 0.0'
```

Parameters altitude (*bool*) – Whether to include `altitude` value. By default it is automatically included if it is non-zero.

format_unicode(*altitude=None*)

format() with pretty unicode chars for degrees, minutes and seconds:

```
>>> p = Point(41.5, -81.0, 12.3)
>>> p.format_unicode()
'41° 30 0 N, 81° 0 0 W, 12.3km'
```

New in version 1.23.0.

Parameters **`altitude`** (*`bool`*) – Whether to include `altitude` value. By default it is automatically included if it is non-zero.

`classmethod from_point`(*point*)

Create and return a new `Point` instance from another `Point` instance.

`classmethod from_sequence`(*seq*)

Create and return a new `Point` instance from any iterable with 2 to 3 elements. The elements, if present, must be latitude, longitude, and altitude, respectively.

`classmethod from_string`(*string*)

Create and return a `Point` instance from a string containing latitude and longitude, and optionally, altitude.

Latitude and longitude must be in degrees and may be in decimal form or indicate arcminutes and arcseconds (labeled with Unicode prime and double prime, ASCII quote and double quote or ‘m’ and ‘s’). The degree symbol is optional and may be included after the decimal places (in decimal form) and before the arcminutes and arcseconds otherwise. Coordinates given from south and west (indicated by S and W suffixes) will be converted to north and east by switching their signs. If no (or partial) cardinal directions are given, north and east are the assumed directions. Latitude and longitude must be separated by at least whitespace, a comma, or a semicolon (each with optional surrounding whitespace).

Altitude, if supplied, must be a decimal number with given units. The following unit abbrevations (case-insensitive) are supported:

- `km` (kilometers)
- `m` (meters)
- `mi` (miles)
- `ft` (feet)
- `nm`, `nmi` (nautical miles)

Some example strings that will work include:

- `41.5;-81.0`
- `41.5,-81.0`
- `41.5 -81.0`
- `41.5 N -81.0 W`
- `-41.5 S;81.0 E`
- `23 26m 22s N 23 27m 30s E`
- `23 26' 22" N 23 27' 30" E`
- `UT: N 39°20' 0'' / W 74°35' 0''`

`classmethod parse_altitude`(*distance*, *unit*)

Parse altitude managing units conversion:

```
>>> Point.parse_altitude(712, 'm')
0.712
>>> Point.parse_altitude(712, 'km')
712.0
>>> Point.parse_altitude(712, 'mi')
1145.852928
```

Parameters

- **`distance`** (*`float`*) – Numeric value of altitude.
- **`unit`** (*`str`*) – `distance` unit. Supported units are listed in *`from_string()`* doc.

`classmethod parse_degrees`(*degrees*, *arcminutes*, *arcseconds*, *direction=None*)
Convert degrees, minutes, seconds and direction (N, S, E, W) to a single degrees number.

Return type float

`class` `geopy.timezone.`**`Timezone`**(*pytz_timezone*, *raw=None*)
Contains a parsed response for a timezone request, which is implemented in few geocoders which provide such lookups.

New in version 1.18.0.

`pytz_timezone`
pytz timezone instance.

Return type `pytz.tzinfo.BaseTzInfo`

`raw`
Timezone's raw, unparsed geocoder response. For details on this, consult the service's documentation.

Return type dict or None

CHAPTER 6

Units Conversion

`geopy.units` module provides utility functions for performing angle and distance unit conversions.

Some shortly named aliases are provided for convenience (e.g. `km()` is an alias for `kilometers()`).

`geopy.units.`**`arcmin`**`(`*degrees=0*, *radians=0*, *arcseconds=0*`)`
    Convert angle to arcminutes.

`geopy.units.`**`arcminutes`**`(`*degrees=0*, *radians=0*, *arcseconds=0*`)`
    Convert angle to arcminutes.

`geopy.units.`**`arcsec`**`(`*degrees=0*, *radians=0*, *arcminutes=0*`)`
    Convert angle to arcseconds.

`geopy.units.`**`arcseconds`**`(`*degrees=0*, *radians=0*, *arcminutes=0*`)`
    Convert angle to arcseconds.

`geopy.units.`**`degrees`**`(`*radians=0*, *arcminutes=0*, *arcseconds=0*`)`
    Convert angle to degrees.

`geopy.units.`**`feet`**`(`*kilometers=0*, *meters=0*, *miles=0*, *nautical=0*`)`
    Convert distance to feet.

`geopy.units.`**`ft`**`(`*kilometers=0*, *meters=0*, *miles=0*, *nautical=0*`)`
    Convert distance to feet.

`geopy.units.`**`kilometers`**`(`*meters=0*, *miles=0*, *feet=0*, *nautical=0*`)`
    Convert distance to kilometers.

`geopy.units.`**`km`**`(`*meters=0*, *miles=0*, *feet=0*, *nautical=0*`)`
    Convert distance to kilometers.

`geopy.units.`**`m`**`(`*kilometers=0*, *miles=0*, *feet=0*, *nautical=0*`)`
    Convert distance to meters.

`geopy.units.`**`meters`**`(`*kilometers=0*, *miles=0*, *feet=0*, *nautical=0*`)`
    Convert distance to meters.

`geopy.units.`**`mi`**`(`*kilometers=0*, *meters=0*, *feet=0*, *nautical=0*`)`
    Convert distance to miles.

`geopy.units.`**`miles`**(*kilometers=0*, *meters=0*, *feet=0*, *nautical=0*)
Convert distance to miles.

`geopy.units.`**`nautical`**(*kilometers=0*, *meters=0*, *miles=0*, *feet=0*)
Convert distance to nautical miles.

`geopy.units.`**`nm`**(*kilometers=0*, *meters=0*, *miles=0*, *feet=0*)
Convert distance to nautical miles.

`geopy.units.`**`rad`**(*degrees=0*, *arcminutes=0*, *arcseconds=0*)
Convert angle to radians.

`geopy.units.`**`radians`**(*degrees=0*, *arcminutes=0*, *arcseconds=0*)
Convert angle to radians.

CHAPTER 7

Exceptions

class geopy.exc.**GeopyError**

Bases: Exception

Geopy-specific exceptions are all inherited from GeopyError.

class geopy.exc.**ConfigurationError**

Bases: *geopy.exc.GeopyError*

When instantiating a geocoder, the arguments given were invalid. See the documentation of each geocoder's `__init__` for more details.

class geopy.exc.**GeocoderServiceError**

Bases: *geopy.exc.GeopyError*

There was an exception caused when calling the remote geocoding service, and no more specific exception could be raised by geopy. When calling geocoders' `geocode` or *reverse* methods, this is the most generic exception that can be raised, and any non-geopy exception will be caught and turned into this. The exception's message will be that of the original exception.

class geopy.exc.**GeocoderQueryError**

Bases: *geopy.exc.GeocoderServiceError*

Either geopy detected input that would cause a request to fail, or a request was made and the remote geocoding service responded that the request was bad.

class geopy.exc.**GeocoderQuotaExceeded**

Bases: *geopy.exc.GeocoderServiceError*

The remote geocoding service refused to fulfill the request because the client has used its quota.

class geopy.exc.**GeocoderAuthenticationFailure**

Bases: *geopy.exc.GeocoderServiceError*

The remote geocoding service rejected the API key or account credentials this geocoder was instantiated with.

class geopy.exc.**GeocoderInsufficientPrivileges**

Bases: *geopy.exc.GeocoderServiceError*

The remote geocoding service refused to fulfill a request using the account credentials given.

class geopy.exc.**GeocoderTimedOut**

Bases: *geopy.exc.GeocoderServiceError*

The call to the geocoding service was aborted because no response has been received within the `timeout` argument of either the geocoding class or, if specified, the method call. Some services are just consistently slow, and a higher timeout may be needed to use them.

class geopy.exc.**GeocoderUnavailable**

Bases: *geopy.exc.GeocoderServiceError*

Either it was not possible to establish a connection to the remote geocoding service, or the service responded with a code indicating it was unavailable.

class geopy.exc.**GeocoderParseError**

Bases: *geopy.exc.GeocoderServiceError*

Geopy could not parse the service's response. This is probably due to a bug in geopy.

class geopy.exc.**GeocoderNotFound**

Bases: *geopy.exc.GeopyError*

Caller requested the geocoder matching a string, e.g., `"google"` > `GoogleV3`, but no geocoder could be found.

CHAPTER 8

Logging

geopy will log geocoding URLs with a logger name `geopy` at level *DEBUG*, and for some geocoders, these URLs will include authentication information.

HTTP bodies of responses with unsuccessful status codes are logged with *INFO* level.

Default logging level is *NOTSET*, which delegates the messages processing to the root logger. See docs for `logging.Logger.setLevel()` for more information.

CHAPTER 9

Semver

geopy attempts to follow semantic versioning, however some breaking changes are still being made in minor releases, such as:

- Backwards-incompatible changes of the undocumented API. This shouldn't affect anyone, unless they extend geocoder classes or use undocumented features or monkey-patch anything. If you believe that something is missing in geopy, please consider opening an issue or providing a patch/PR instead of hacking around geopy.
- Geocoding services sometimes introduce new APIs and deprecate the previous ones. We try to upgrade without breaking the geocoder's API interface, but the `geopy.location.Location.raw` value might change in a backwards-incompatible way.
- Behavior for invalid input and peculiar edge cases might be altered. For example, `geopy.point.Point` instances previously did coordinate values normalization, though it's not documented, and it was completely wrong for the latitudes outside the *[-90; 90]* range. So instead of using an incorrectly normalized value for latitude, a `ValueError` exception is now thrown (#294).

Features and usages being phased out are covered with deprecation `warnings` when possible. Make sure to run your python with the `-Wd` switch to see if your code emits the warnings.

To make the upgrade less painful, please read the changelog before upgrading.

CHAPTER 10

Changelog

Changelog for 1.x.x series.

For changes in the 0.9 series, see the 0.9x changelog.

CHAPTER 11

Indices and search

- genindex
- search

Python Module Index

Index