

---

# **python-geoip documentation**

***Release 1.0***

**Armin Ronacher**

November 27, 2015



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>API</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



python-geoip is a library that provides access to GeoIP databases. Currently it only supports accessing MaxMind databases. It's similar to other GeoIP libraries but comes under the very liberal BSD license and also provides an extra library that optionally ships a recent version of the Geolite2 database as provided by MaxMind.

The `python-geoip-geolite2` package includes GeoLite2 data created by MaxMind, available from [maxmind.com](https://maxmind.com) under the Creative Commons Attribution-ShareAlike 3.0 Unported License.



---

# Installation

---

You can get the library directly from PyPI:

```
pip install python-geoip
```

If you also want the free MaxMind Geolite2 database you can in addition:

```
pip install python-geoip-geolite2
```





---

## Usage

---

If you have installed the `python-geoip-geolite2` package you can start using the GeoIP database right away:

```
>>> from geoip import geolite2
>>> match = geolite2.lookup('17.0.0.1')
>>> match is not None
True
>>> match.country
'US'
>>> match.continent
'NA'
>>> match.timezone
'America/Los_Angeles'
>>> match.subdivisions
frozenset(['CA'])
```

If you want to use your own MaxMind database (for instance because you paid for the commercial version) you can open the database yourself:

```
>>> from geoip import open_database
>>> db = open_database('path/to/my.mmdb')
```



## API

`geoip.open_database(filename)`

Open a given database. This currently only supports maxmind databases (mmdb). If the file cannot be opened an `IOError` is raised.

`geoip.geolite2 = <PackagedDatabase 'geolite2'>`

Provides access to the geolite2 cities database. In order to use this database the `python-geoip-geolite2` package needs to be installed.

`class geoip.Database`

Provides access to a GeoIP database. This is an abstract class that is implemented by different providers. The `open_database()` function can be used to open a MaxMind database.

Example usage:

```
from geoip import open_database

with open_database('data/GeoLite2-City.mmdb') as db:
    match = db.lookup_mine()
    print 'My IP info:', match
```

`close()`

Closes the database. The whole object can also be used as a context manager. Databases that are packaged up (such as the `geolite2` database) do not need to be closed.

`get_info()`

Returns an info object about the database. This can be used to check for the build date of the database or what provides the GeoIP data.

**Return type** `DatabaseInfo`

`get_metadata()`

Return the metadata dictionary of the loaded database. This dictionary is specific to the database provider.

`lookup(ip_addr)`

Looks up the IP information in the database and returns a `IPInfo`. If it does not exist, `None` is returned. What IP addresses are supported is specific to the GeoIP provider.

**Return type** `IPInfo`

`lookup_mine()`

Looks up the computer's IP by asking a web service and then checks the database for a match.

**Return type** `IPInfo`

`class geoip.IPInfo(ip, data)`

Provides information about the located IP as returned by `Database.lookup()`.

**continent**

The continent as ISO code if available.

**country**

The country code as ISO code if available.

**get\_info\_dict()**

Returns the internal info dictionary. For a maxmind database this is the metadata dictionary.

**ip**

The IP that was looked up.

**location**

The location as (lat, long) tuple if available.

**subdivisions**

The subdivisions as a list of ISO codes as an immutable set.

**timezone**

The timezone if available as tzinfo name.

**to\_dict()**

A dict representation of the available information. This is a dictionary with the same keys as the attributes of this object.

**class** `geoiip.DatabaseInfo`

Provides information about the GeoIP database.

**date = None**

Optionally the build date of the database as datetime object.

**filename = None**

If available the filename which backs the database.

**internal\_name = None**

Optionally the internal name of the database.

**provider = None**

Optionally the name of the database provider.

## g

geoip, 3



## C

`close()` (geoip.Database method), 7  
`continent` (geoip.IPInfo attribute), 7  
`country` (geoip.IPInfo attribute), 8

## D

`Database` (class in geoip), 7  
`DatabaseInfo` (class in geoip), 8  
`date` (geoip.DatabaseInfo attribute), 8

## F

`filename` (geoip.DatabaseInfo attribute), 8

## G

`geoip` (module), 1  
`geolite2` (in module geoip), 7  
`get_info()` (geoip.Database method), 7  
`get_info_dict()` (geoip.IPInfo method), 8  
`get_metadata()` (geoip.Database method), 7

## I

`internal_name` (geoip.DatabaseInfo attribute), 8  
`ip` (geoip.IPInfo attribute), 8  
`IPInfo` (class in geoip), 7

## L

`location` (geoip.IPInfo attribute), 8  
`lookup()` (geoip.Database method), 7  
`lookup_mine()` (geoip.Database method), 7

## O

`open_database()` (in module geoip), 7

## P

`provider` (geoip.DatabaseInfo attribute), 8

## S

`subdivisions` (geoip.IPInfo attribute), 8

## T

`timezone` (geoip.IPInfo attribute), 8  
`to_dict()` (geoip.IPInfo method), 8